UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

CONTRIBUTIONS TO THE SOLUTION OF THE ENERGY EFFICIENT FILE DISTRIBUTION PROBLEM

Autor: Kshitiz Verma
Director: Dr. Alberto García Martínez

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés (Madrid), septiembre de 2016

UNIVERSIDAD CARLOS III DE MADRID

PH.D. THESIS

CONTRIBUTIONS TO THE SOLUTION OF THE ENERGY EFFICIENT FILE DISTRIBUTION PROBLEM

Autor: Kshitiz Verma
Director: Dr. Alberto García Martínez

DEPARTMENT OF TELEMATIC ENGINEERING

Leganés (Madrid), September 2016

TESIS DOCTORAL

Contributions to the solution of the energy efficient file distribution problem

Autor: Kshitiz Verma
Director: Dr. Alberto García Martínez

Firma del tribunal calificador:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

Acknowledgements

I will begin it by writing that I would never have started writing it unless I had Alberto as my supervisor. I am grateful to have him as my supervisor. He has been a continuous support, morally and technically, always willing to hear from me and then responding quickly. His responses have made this Thesis a reality that otherwise was simply not possible. I would like to thank him from the depth of my heart for being there with me when no one else believed in me and continuing that support for years.

I would also like to thank Prof. Sadanand Gokhale, Prof. Ravi Gorthi of LNMIIT for their continuous support and motivation to the extent that I can safely say that this Thesis was partly funded by the LNMIIT, Jaipur. Many thanks for their accommodativeness for taking time out of their busy schedule and sometimes even going out of their way to help me out.

I would like to thank my collaborators Antonio Fernández Anta, Gianluca Rizzo, Shmuel Zaks, Ruben Cuevas and Arturo Azcorra for their insightful discussions and directions. In particular, I would like to thank Antonio, Gianluca and Shmuel to help me retain a positive attitude throughout the course of the Thesis.

Thanks to Pradeep, Syed and Andres for making my stay in Madrid full of joy. I would also like to thank them for uncountable favours that they have done without asking for anything in return. You guys rock.

I am thankful to all family members, when I say so, I mean the family of my grandparents. Everyone has played a non-trivial role that has helped me finish this task. I would particularly like to thank my father *Shri* Mohan Lal and mother *Smt* Kamla for always allowing me to do whatever I wanted to do and supporting my decisions with the highest regard. They have always been an inspiration to me and will always be. I owe thanks to them in many ways but I will settle it with just this due to lack of space.

I would also like to express my gratitude towards my one year old son Tinchu, who used to stay awake all night to check if I am still working. Finally, a sincere thanks goes to my wife Jyoti, she is the one who has suffered the most because I wanted to write this document. Her selfless love, motivation and allowance to pursue whatever was helping me in finishing my Thesis has been incredible. Thank you for always being there.

En mi segunda pais Esapña, mi segunda madre Isabel han ayudado mucho por alli.

Han planchado mis camisetas, preparado las comidas y cenas, para enseñar Español (con error o sin error, pero he escrito yo, sin consultar a nadie, google tampoco), has cuidado por mi como mi madre y muchisima mas cosas. Gracias para todo Isabel. Besos. Lo pido muchisima gracias a Mari tambien. Tambien has hecho cosas para mi, creo que no vas a hacer para nadie. Gracias los dos.

Abstract

It has been realized that energy, one of the key requirements for modern human civilization, must be used efficiently for the civilization to be sustainable. The Information and Communications Technology (ICT) sector is no exception. It has been shown through research that ICT is consuming energy comparable to the aviation sector and is still increasing rapidly. In order to address this issue, many energy efficient approaches applicable to ICT sector have been proposed in the literature.

In this Thesis, we pick one of the most ubiquitous task in ICT, *file distribution* and concentrate on finding ways of transferring a file from one server to many hosts in the most energy efficient manner. We study the problem for one server and many host problem but our algorithms can be applied to many general scenarios including P2P file distribution, replication of content in a cloud, synchronization of caches in content distribution networks, downloading software updates to millions of PCs worldwide, and many more applications where the data disseminated does not have to be consumed instantaneously, for example, in video streaming.

We study the problem for one server and many hosts but our algorithms can be applied to more general scenarios including P2P file distribution, replication of content in a cloud, synchronization of caches in content distribution networks, downloading software updates to millions of PCs worldwide, We assume that the time is slotted and that the file is transfered in units of data called blocks. Each host can have arbitrary power consumption, upload and download capacities. To begin with, we prove that the problem of energy efficient file distribution is NP-complete. In order to solve the problem optimally, we assume additional constraints and impose that all the hosts involved in the file transfer should have same upload and download capacity. Moreover, we also assume that the upload and download capacities are such that they are integral multiples of each other, which is typically the case. Under these conditions, we prove lower bounds on energy and design algorithms for file distribution that achieve the calculated lower bounds. Our algorithms minimize the amount of time a host has to be on to download and/or upload in the distribution process.

Apart from being theoretically sound, we also evaluate our model by extending our analysis through extensive numerical evaluation to compare the proposed algorithms

with the already existing schemes of transfers. Our algorithms show promising improvement over not just the traditional energy agnostic approaches but also over the schemes designed for energy efficient file distribution. It has been shown that our algorithms are at least 50% more energy efficient than any of the proposals compared with. We advance our numerical analysis to relax the constraints in the theoretical analysis and conclude that our algorithms are also applicable in scenarios in which the computing and networking hardware is energy efficient. Our algorithms can exploit the power proportionality of the devices.

No efficiency comes without a cost. In this case, we pay the cost in terms of the tight synchronization that our algorithms require. However, we argue that such a tight synchronization at each slot level is possible in today's Internet particularly if the algorithms are applied to the hosts inside a corporation in which all the hosts and network are controlled by a central entity. For example, servers of a cloud, content distribution network, software updates inside a corporation, etc.

Table of Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

Concerns over the growing energy demands of humankind are increasing and energy efficiency is reported to be one of the major technological challenges of our times [1]. The environmental problems related to Green House Gases have shown significant increase recently and the measures to counter its impact is reflected in policies of the governments [2–4] and organizations [5–7]. Apart from environmental effects, increasing energy demand has negative impacts on economy too.

As far as the enormous energy consumption is concerned, the Internet and associated computing and networking devices of Information and Communication Technologies (ICT) are no exception [8]. Its carbon footprint is comparable to that of the aviation sector [9]. The problem was first identified in [10] [11] [12] . Since then, energy efficiency has become an essential metric in the design of computing and networking device components. According to various observations like [13], the Internet is expected to account for roughly 2-3% of the total global energy consumption but it can be as high as 10% for a developed nation like the United Kingdom. Even though it has been shown that greening the Internet is not easy [65], the researchers have provided different energy efficient solutions for various branches of ICT including, but not limited to the hardware design, protocol design, topology considerations, data center design, energy efficient content distribution and file sharing.

Among many areas of ICT in which energy efficiency can be developed, in this Thesis, we focus on a special but ubiquitous process in communication technologies, file distribution. We concentrate on energy consumption of the process of distributing a file from one to many hosts in different scenarios. File-sharing applications are usually run by PCs or laptops. We consider all the cases in which no two hosts are in the same collision domain, i.e., for example, if a host is connected via wireless then it is assumed that there are no collisions with the frames of the host. In addition, operations such as software updates can be defined also as file distribution processes. The updates are released and all the hosts for which the update is relevant are strongly recommended to download it [14]. If a user uploads a file to a cloud, it is replicated in many servers of the cloud [15–18]. All these kind of data transfer can be classified as file distribution from one to many. The same holds for the updates of caches of a content distribution network, the content need to be replicated in many servers located around the world.

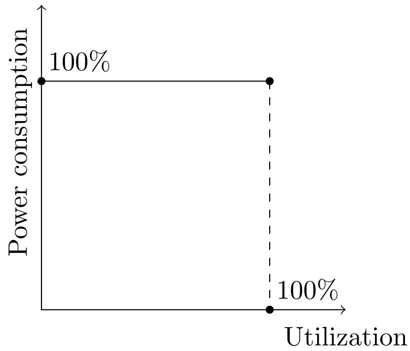
Having understood the meaning of file distribution, in this Thesis, we provide a modeling framework for the problem and prove that the problem is NP-hard. We solve it optimally for some practical cases and provide heuristics whenever possible to address the most general case. Simulations are presented for the cases that are more relevant in practice, showing the performance of the solutions provided. We emphasize that our results are valid even if the networking/computing hardware becomes more energy efficient. Irrespective of the power profile of the devices presented in Fig. 1.1, our algorithms yield substantial energy savings.

In the rest of the chapter, Section 1.1 describes the need and importance of energy efficiency in the ICT sector. Section 1.2 focuses on the phenomenon addressed in the Thesis, energy efficiency in file distribution. Section 1.3 presents the problem and its complexity. Section 1.4 briefs the basic idea behind the algorithms and their optimality. Section 1.5.3 reports some of the important results not covered in theory but are relevant in practice. Finally, organization of the rest of the Thesis is provided in Section 1.6.

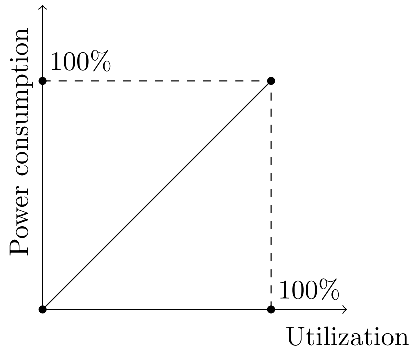
1.1. Drive for a Greener Internet

The design of network devices, protocols, applications or services have not taken energy efficiency into account until very recently. If new energy mechanisms and solutions are not adopted, the energy consumption of the ICT sector is expected to double in the next decade [20]. Data from [21] provides a proof that even though the devices like desktops, laptops, LCDs, CRTs, etc are getting more energy efficient the overall aggregate of the energy consumed by them has increased. The worldwide electricity use for these devices was aggregate 220 TWh/year in 2006 which increased to 307 TWh/year in 2012. Thus, even marginal energy savings at the end terminals will have a huge impact on global energy savings.

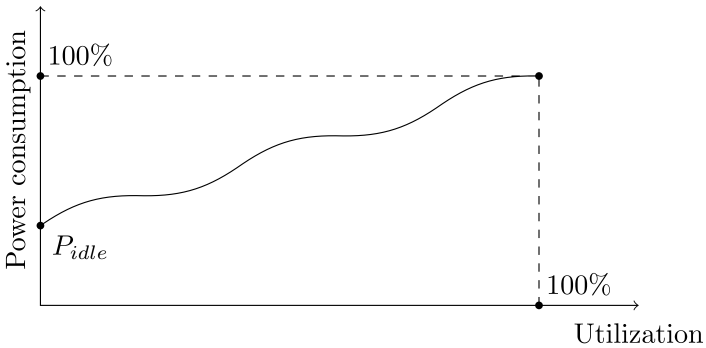
To minimize the impact of ICT on environment, as well as to mitigate the impact of increasing energy costs [22], energy efficiency has been addressed in network design in many ways. Fig. 1.1 presents three different energy profiles of devices. Before power was identified as a problem, the energy profiles for most of the components of computer hardware used to be as depicted in Fig. 1.1a. In this profile, there is only one power level, irrespective of the utilization of the device. It is very difficult to achieve the goal of 100% energy proportionality, as shown in Fig. 1.1b. One can, for instance, exploit the existing technology that supports low power modes or switching off the devices whenever it is possible [23], [24] or use the newly developed energy efficient devices to design energy



(a) Energy agnostic devices consume the same power once switched on.



(b) Ideal power aware devices consume as per their utilization.



(c) The current state of the energy profiles of many devices [19]

Figure 1.1: Different power profiles for networking/computing equipments like PCs, routers, switches, etc.

aware networks with performance. These techniques can be used to approach energy proportionality in end systems, i.e., making the power consumed proportional to the level of CPU or network activity, as opposed to the current constant power consumption irrespective of its utilization.

The efforts of researchers in the last decade have made it possible for the devices to have a power profile as shown in Fig. 1.1c, which can be seen to lie somewhere between the two extremes of having one and infinite power levels. There are different ranges of utilization which correspond to a certain power level. From now on, from energy efficient devices, we mean the profiles corresponding to Fig. 1.1c as discussed in [19]. However, energy proportionality of the different elements alone is not enough to reduce the overall energy wastage in most distributed systems. It needs to be complemented by a redesign of the services (e.g., file sharing, web browsing, etc.) in a way that optimizes the utilization of hosts and network resources from energy perspective. It means that the higher layers should be designed to exploit energy efficient hardware in an optimized manner. If an application behaves such that it never lets the hardware go to sleep mode, then this is equivalent to not having sleep modes at all.

In order to save energy at the end hosts, cooperation and collaboration is needed from the end hosts. [25] already provides insights about huge potential for energy savings in the last mile, if the end users are willing to cooperate. More energy savings can be obtained if the tasks that we focus are very common for the end hosts. In this Thesis, we focus on a ubiquitous task in networking, file distribution, and its huge potential in addressing energy efficiency, as discussed in the next Section.

1.2. Energy Efficiency in File Distribution

It is important to understand that we do not address energy efficiency in all kinds of data broadcast. In particular, our approach works well with the applications in which the file is downloaded by the hosts for a later use. For example, our work cannot be directly applied to video streaming, or if an email is to be sent to many other persons. By file we mean a piece of data that is to be sent to many users and once received the whole file, then only users need it. Many relevant applications have already been talked about before.

Fig. 1.2 shows an instance of the problem scenario that we consider in the Thesis. We also assume that all the hosts are reachable from each other and we do not consider the energy consumed by the intermediate devices like router, access points, switches, etc. because we assume that they cannot be turned off. As long as the hosts (like A, B and C) are connected via a wireless connection but not in the same wireless collision domain (same access point, for example), our algorithms for file distribution can be used. We put no restrictions on the number of hosts that can participate in the file distribution but we consider 500-1500 as a normal scenario. We also assume that the files are at least a few megabytes in size. There is no upper limit on the file size.

It has been observed that most of the end terminals involved in file sharing are known to be on just to download and upload files [26], i.e., they can be assumed to be doing nothing but downloading and uploading files. Hence, huge energy savings are possible if files are transfered in energy efficient manner so that the hosts can be switched off as soon as the file transfers are completed. As demonstrated by previous studies like [27], homes and organizations (i.e., end-hosts) are responsible for 75% of the overall Internet energy consumption, whereas networking devices (e.g., routers) and data-centers are responsible for the other 25%. The existing file distribution services, such as peer-to-peer (P2P) file sharing, one-click-hosting (OCH), software release, etc., represent a major fraction of current Internet traffic, ranging between 18 and 30 percent [14], [28], [29]. The

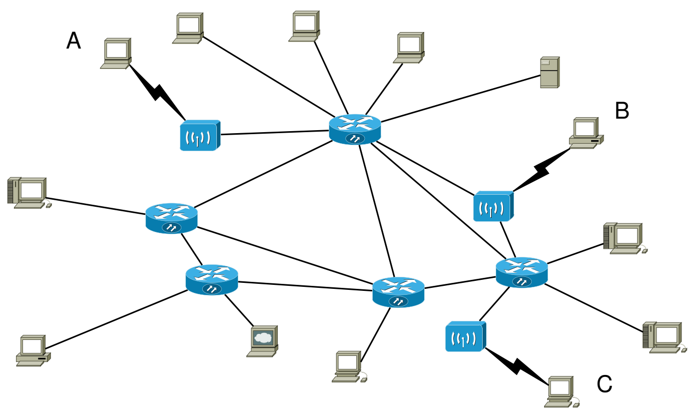


Figure 1.2: A typical scenario that we consider. Notice that hosts A, B and C are connected via a wireless connection.

combined effect of the two previous arguments suggest that the file-sharing applications are responsible for a significant portion of the overall energy consumption in the Internet.

The focus of this Thesis, energy efficiency in file distribution is subject of the research in [30], [31] and [32]. This research defines a problem close to our formulation, i.e., one server has to disseminate a file to n hosts in an energy efficient manner. However, there are many subtle differences. In [30], the analysis is mainly dependent on [33] and works only for a network with three hosts. For $n > 3$, they provide only simulations. In this context, [31] does much better job of defining a family of near-optimal strategies, but under a fluid limit model, in which the file is split into infinitesimally small blocks. For this reason, their results cannot be extended to practical settings, where block sizes must be lower bounded to keep bounded the amount of extra transmissions (and extra energy spent) due to control data (protocol overheads, etc). As we show in our Thesis (see Section 4.4), the dependence on the blocks size and number of the energy consumption of a distribution scheme is non negligible in any practical scenario. Their approach also requires existence of some low power hosts (like notepads) that have high upload capacity. The aggregate of the server and such low power devices essentially make a hypothetical server that has the power equivalent to all the low power devices and the server and upload capacity is the sum of all the upload capacities. This set of hosts start uploading blocks to the hosts at their maximum download speed. As in their algorithms a subset of hosts (which always contains at least the server) stays on for the whole duration of the scheme, the total energy consumption of the proposed algorithms is higher with respect to the optimal values (that we define here) by at least a factor directly proportional to the power consumed by the server and to the makespan of the distribution scheme. As we show later in the Thesis, for such schemes the total energy consumption is up to twice that of the optimal schemes we propose, depending on the specific settings.

Our investigation aims at modeling, analyzing and evaluating the performance of energy-efficient file distribution algorithms in a controlled collaborative environment without assuming existence of any low power hosts or fluid model. As already discussed before, achieving energy efficiency at the end hosts is a non-trivial task. The price that we pay in our algorithms is a tight synchronization of all the hosts participating in the file distribution process. However, such tight synchronization is achievable in today's Internet, at least for all the hosts that are under the control of the same administrative domain, e.g., a big corporate, data centers, content distribution networks, campus LAN, etc. We also evaluate our schemes through simulations for a more general scenario.

1.3. Energy Efficient File Distribution and its Complexity

In Chapter 3, we model the file distribution process, such that a file consisting of β blocks, initially available only at a server S is to be distributed to n hosts $\{0, 1, \cdots, n-1\}$ as an optimization problem to minimize the energy consumed. Each host i has upload capacity u_i and download capacity d_i. We first state the network and system energy model and our basic assumptions. Then, we reduce the problem to the partition problem to prove that the problem is NP-hard if all the hosts have different upload and download capacities, hosts are allowed to upload to as many users as they want at any upload speed. We change the conditions and show two other variants of the problem to be NP-complete as well. Hence, it is not clearly understood, which are the parameters that we should relax in order to solve the problem optimally in polynomial time.

To understand the problem complexity better, we define the notion of a *state* to capture the status of completion of download at each host. The initial state is defined as empty state in which no host, has downloaded the file. When the process ends, the final states are the ones in which all the hosts have downloaded the complete file. Energy is spent in order to reach to current state from the states preceding it. The difference between each final state is the amount of energy required to achieve it. An optimal final state is the one that consumes the minimum energy.

From this method, we infer that one of the ways to solve the problem optimally is to require that the data transfers are done in slots of time. In order to ensure a fixed slot length for all the transfers, the file should be divided into equal sized blocks and all the hosts should have the same upload or download capacities, i.e., $u_i = u \; \forall i \in \{S, 0, 1, \cdots, n-1\}$ and $d_i = d \; \forall i \in \{S, 0, 1, \cdots, n-1\}$. We elaborate on these situations in the next section.

1.4. Energy Optimal Schemes for Restricted Cases

In this section, we discuss the results presented as schemes in the Thesis. We divide our results in three basic scenarios depending on the relationship between the upload (u) and download (d) capacities that are discussed next. First, we present the results for the case in which $d = u$ (Chapter 4). Secondly, the case for which $u = kd$ for an integer $k > 1$ (Chapter 5). Finally, we present the results for case $d = ku$ for an integer $k > 1$ (Chapter 6). This Thesis provides proven optimal schemes for the first two cases and we conjecture that the algorithms for the final case are optimal too.

As stated before, time can be divided into slots. Hence, we can visualize a scheme that accomplishes file distribution as a set of transfers that are carried over from the first slot until the last slot in which all the hosts receive the whole file. The main intuition behind the proposed file distribution algorithm is to activate node uploading in the same slots as downloading occurs. In this way, we try to reduce the high amount of energy spent when keeping a node turned on just for either downloading or uploading, but not both.

The schemes provided in this Thesis keep hosts on in such a manner that the total time each host is on to download as well as upload a file is minimized. Even though it is not a time optimization problem, because of the above properties, the schemes still finish in $O(n + \beta)$.

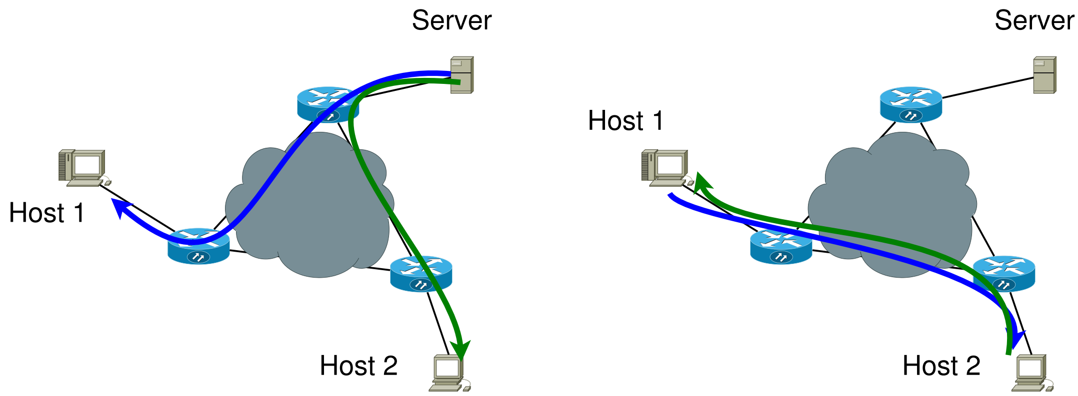
1.4.1. Download = Upload Capacity

In order to understand the problem, we start with the simplest case, i.e., $d = u$. Note that we put no restriction on the power consumption of the hosts, i.e. power consumed by host i is P_i. This seemingly restricted case helps us find solutions to the more involved cases too. Fig. 1.3 shows how the scheme works. Note that in all the slots only those hosts that are involved in block transfer are switched on.

Intuitively, we can see that this strategy leads to the devices turned on for minimum possible time to upload and download. Thus, consuming minimum possible energy. Also note that irrespective of the power consumption of the hosts, each host must be on for at least 2 slots to download or upload the file.

1.4.2. Upload $>$ Download Capacity

Fig. 1.4 demonstrates how the scheme will work in this case through a particular example of $u = 2d$. In genral, if $u = kd$, then the server can upload simultaneously to k hosts, further increasing the energy efficiency. Note that even in this case, the power consumption of the host i is P_i. As we can see there is a very little difference with respect to the energy consumption in the previous scenario. The energy that is saved in this case is during the transfer made by the server because it can upload both the blocks to the two hosts in just one slot. The server can go to sleep just after one slot. In the example, it can be seen that $u > d$ is used by the server only as the maximum upload speed can



(a) In the first slot, the server uploads the first block to Host 1 and in the next slot, the server uploads second block of the file to Host 2. The server can go to sleep mode now.

(b) Both the hosts are active to download the remaining block of the file from each other. They go to sleep after receiving the two blocks.

Figure 1.3: The download capacity (d) of all the hosts are equal to the upload capacity (u) of all the hosts, i.e., $u_1 = u_2 = u_S = u$ and $d_1 = d_2 = d_S = d$ and that $d = u$. The file is divided into 2 blocks. Power consumed by server is P_S, host i is $P_i, i \in \{1, 2\}$.

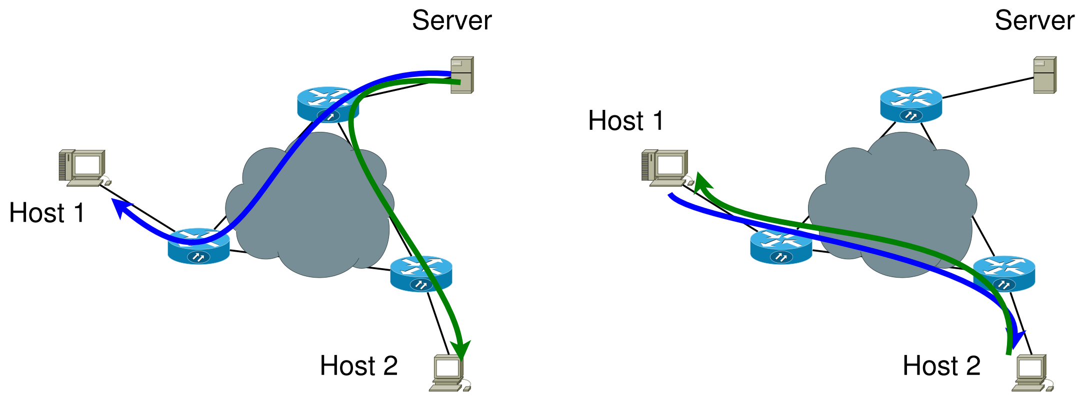
be d because it is the maximum speed a host can download. So even though upload at a higher rate is possible, it is not permitted by the download capacity of the hosts. Again it must be noted that only those hosts who are involved in a block transfer are switched on. Also note that irrespective of the power consumption of the hosts, each downloading host must be on for at least 2 slots. However, the server has to be on for only one slot to upload the file.

1.4.3. Download $>$ Upload Capacity

We turn our attention to the case for which download capacity is an integral multiple of the upload capacity, i.e, $d = ku$. In particular, we consider $d = 2u$ for the examples presented in this section. In this particular case, the relationship between the power consumption of the hosts become important. Hence, we divide this case further in two: *Energy Homogeneous System* and *Energy Heterogeneous System*. In *energy-homogeneous system*, all the hosts have the equal power consumption. Whereas in *energy-heterogeneous system*, power consumption of the hosts can be arbitrary.

1.4.3.1. Homogeneous Power

If we assume that all the hosts participating in the file distribution process have equal power consumption as shown in Fig. 1.5. Note that this example has three hosts and one server. The file at the server is divided into three blocks which it sends to the three hosts in three different slots. Since the server has uploaded all the blocks of the file. It can go to sleep mode. After the completion of the three slots in Fig. 1.5a, all the three



(a) Assume $u = 2d$. In the first slot, the server simultaneously uploads the first block to Host 1 and the second block to Host 2. The server can go to sleep now.

(b) Both the hosts are active to download the remaining block of the file from each other. They go to sleep after receiving the two blocks.

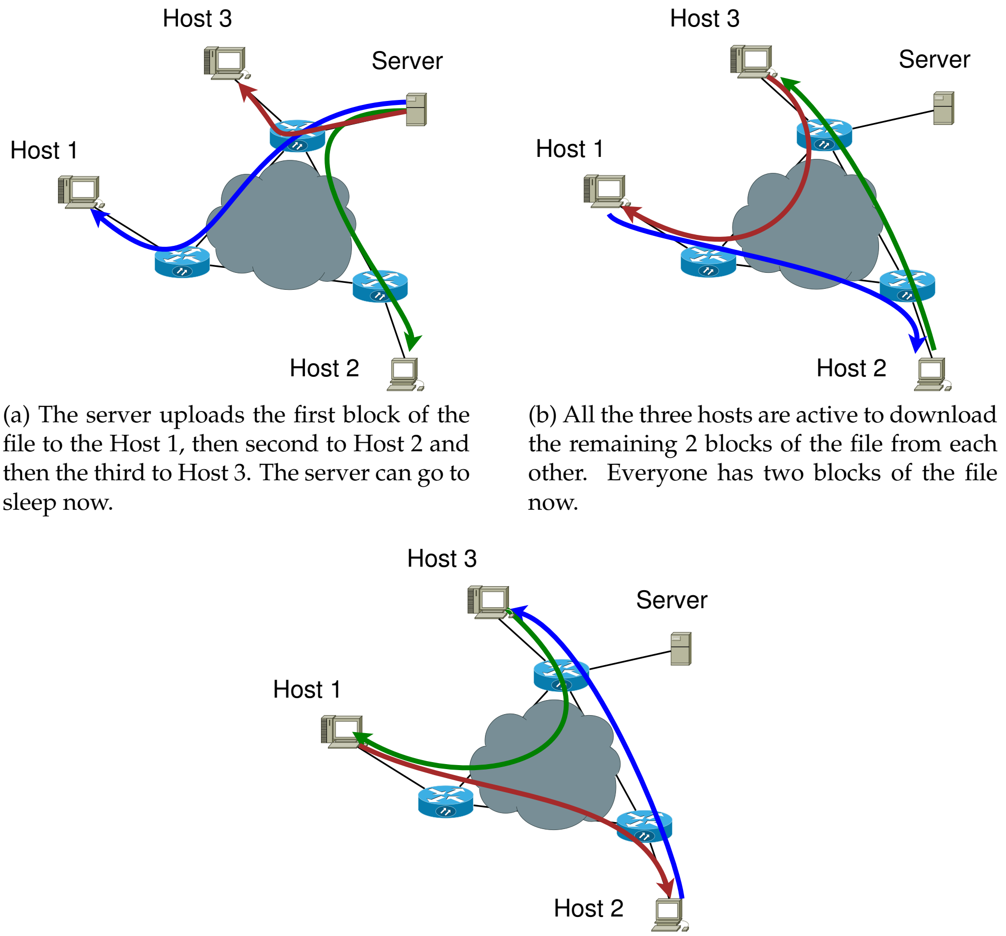
Figure 1.4: The download capacity (d) of all the hosts are equal to the upload capacity (u) of all the hosts, i.e., $u_1 = u_2 = u_S = u$ and $d_1 = d_2 = d_S = d$ and that $u = 2d$. The file is divided into 2 blocks. Power consumed by server is P_S, host i is $P_i, i \in \{1, 2\}$.

hosts have exactly one block of the file. They just need to send these blocks to each other to complete file distribution. In the remaining two figures, Fig. 1.5b and Fig. 1.5c, they just upload and download from each other. Note that host 3 is participating in the distribution process from the beginning. Once they have all the blocks of the file they can go to sleep mode.

For energy-homogeneous case, we also prove that having $d = 2u$ is optimal. $k > 2$ does not help in reducing energy consumption, i.e., even if we have for example $d = 10u$, still it is feasible to use $d = 2u$ and achieve maximum energy savings.

1.4.3.2. Heterogeneous Power

In this case, we demonstrate the importance of the relationship between hosts for optimal schemes. The basic idea is that if there is a host which consumes very high power compared to the other hosts, then such hosts should be served once enough upload capacity is there so that they can download at their maximum rate. Fig. 1.6 shows an example for this scenario. Note that there are three hosts to download the file that is divided into two blocks. In Fig. 1.6a, the server uploads the first block to host 1 in the first slot and the other block to host 2 in the next slot. The server can go to sleep. After that, in Fig. 1.6b the two hosts exchange the blocks with each other. These transfers are ignoring the presence of Host 3. Since the power consumption of Host 3 is very high, it is better to upload all the file to it in the least amount of time so that the energy consumed by it to download the file is minimized. This kind of transfer shows us the importance of relationship between power consumption, which we did not encounter in the earlier

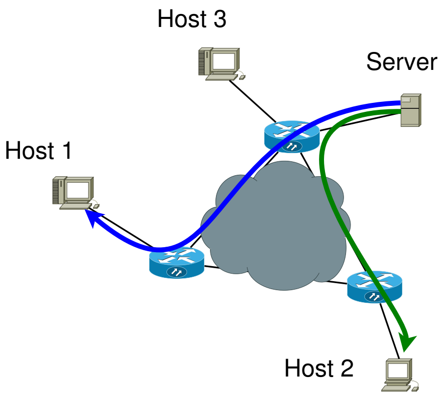


(a) The server uploads the first block of the file to the Host 1, then second to Host 2 and then the third to Host 3. The server can go to sleep now.

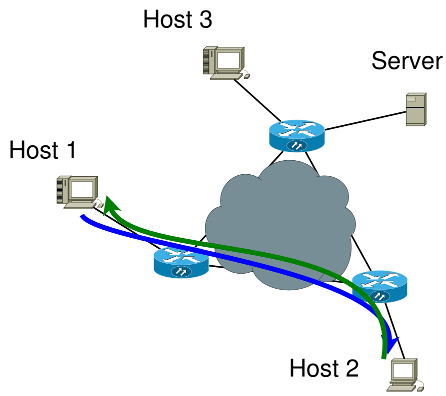
(b) All the three hosts are active to download the remaining 2 blocks of the file from each other. Everyone has two blocks of the file now.

(c) All the three hosts are active to download the remaining block of the file from each other.

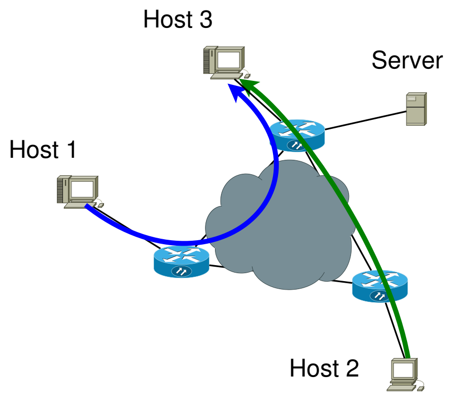
Figure 1.5: The example download capacity (d) of all the hosts are equal to the upload capacity (u) of all the hosts, i.e., $u_1 = u_2 = u_S = u$ and $d_1 = d_2 = d_S = d$ and that $d = 2u$. The file is divided into 3 blocks. Power consumed by server is P_S, host i is $P_i, i \in \{1, 2, 3\}$ and $P_1 = P_2 = P_3 = P_S = P$.



(a) The server uploads first block of the file to Host 1 and then second block of the file to Host 2. The server can go to sleep now.



(b) Both the hosts are active to download the remaining part of the file from each other. Note that the Host 3 still remains inactive.



(c) Since $d = 2u$, Host 1 and Host 2 simultaneously upload to Host 3.

Figure 1.6: The example download capacity (d) of all the hosts are equal to the upload capacity (u) of all the hosts, i.e., $u_1 = u_2 = u_S = u$ and $d_1 = d_2 = d_S = d$ and that $d = 2u$. The file is divided into 2 blocks. Power consumed by server is P_S, host i is $P_i, i \in \{1, 2, 3\}$.

cases.

In Fig. 1.6, if we also assume that $P_3 >> P_1 + P_2$, then it is better for the host 3 to download as much as possible in one slot. This kind of transfers make it different from the schemes that have been presented until now.

In the energy heterogeneous case, having high $\frac{d}{u}$ ratio can help in reducing the energy consumption if there are sufficient hosts that have lower power consumption.

1.5. Summary of Research Contributions

This section summarizes the results presented in the Thesis. We first present the complexity theoretic results, then we present a summary of the algorithms designed in the Thesis and finally we brief the results of the performance analysis.

1.5.1. Complexity Theoretic Results

Table 1.1 summarizes the cases for which the problem has been proven to be NP-complete. We relax some of the assumptions to solve the problem optimally, for which the summary is presented next.

	Upload at any speed	Upload at full speed
Upload to many	NP-Complete	Not studied
Upload to at most one	NP-Complete	NP-Complete

Table 1.1: Complexity theoretic results in Chapter 3.

1.5.2. Algorithmic Results

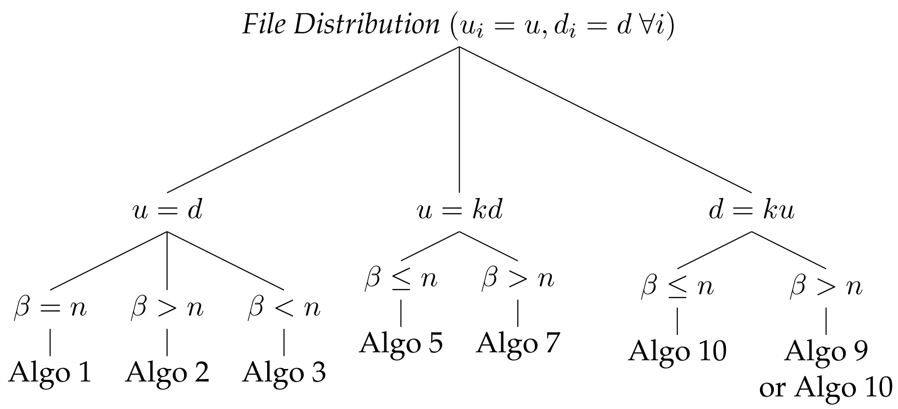


Figure 1.7: Summary of the algorithmic research contributions presented in Chapters 4 - 6 of the Thesis.

Fig. 1.7 summarizes the algorithmic research contributions in the thesis. In this figure and throughout the Thesis, u and d represent upload and download capacities, $k \geq 2$ is an integer, β and n represent the number of blocks and the number of clients respectively. The problem at the root is computationally tractable which is further divided into three cases depending on the relationship between upload and download capacities. Each case is further subdivided depending on the relation between the number of blocks and the number of hosts to download the file.

1.5.3. Performance Analysis

Finally, our empirical evaluation allows us to validate analytical results on the performance of the proposed algorithms. The obtained results support our claim that the proposed algorithms reduce the energy consumption in a file distribution process with respect to any centralized file distribution schemes. In particular, the simulations show that, our collaborative schemes achieve significant energy savings with respect to largely used centralized file distribution systems. These savings range between 50% and two order of magnitude, depending on the centralized scheme under consideration.

We compare our algorithms against the energy consumed by different sequential as well as energy efficient P2P schemes. For a given file size, block size will determine the number of blocks, we evaluate energy savings for different practical block sizes (64KB, 256KB, 1MB, 4MB) and compare the schemes with the aforementioned schemes. We also study the impact of upload capacity u (for example, u=10Kbps, 100Kbps, 1Mbps, 10 Mbps) as well as the download/upload capacity ratio. In the theory, we assume that the hosts don't consume any energy during switch on/off, we evaluate the performance of our schemes when this is not the case. In theory we also assume that the hosts consume the same power whether it is downloading and/or uploading or doing nothing. We relax this assumption and study the impact of load dependence on our algorithms.

1.6. Organization of the Thesis

The rest of the Thesis comprises of seven chapters and is structured as follows. Chapter 2 revises related work and the state of the art in the research relevant to the Thesis. Chapter 3 lays down the assumptions, system, network and energy model along with definitions and terminology used throughout the Thesis. It also formalizes the problem and provides a characterization of the complexity of the problem. Chapter 4 introduces the most basic versions of the optimal algorithms that are designed throughout the course of the Thesis. We prove lower bounds on the energy consumption for this case and provide optimal algorithms achieving that lower bound. We also provide examples of execution of the algorithms on sample scenarios. Chapter 5 considers a more complicated scenario in which upload capacity is an integral multiple of download capacity. The algorithms designed for this case are more sophisticated versions of the algorithms presented in the last chapter but the basic idea remains the same. The extra upload capacity gives more freedom.We prove lower bounds on energy consumption for this case and provide optimal algorithms achieving the bounds. Chapter 6 considers a complementary case to the previous two chapters. We consider the scenario in which download capacity is an integral multiple of the upload capacity. It turns out that this case is the most complicated of all and we provide schemes that we conjecture are optimal. In Chapter 7, we present our simulation study in which we study the performance of our algorithms on various parameters. Our algorithms are capable of saving up to 50% of the energy consumed compared to an energy efficient method making use of proxy in a file distribution process by the end hosts. The savings can go as high as two to three orders of magnitude compared to energy agnostic schemes. Many other cases that are not taken up in the theory are presented. Finally, Chapter 8 concludes the lessons learnt and provides the future directions in which more research is required to understand the problem in a better manner and finally apply them to the file distribution applications. Finally, we also provide an appendix to prove the correctness of the basic algorithms designed in Chapter 4 in the Thesis.

Chapter 2

Background and Related Work

Ever since the problem of energy-avid networking has been identified, researchers have devoted enormous amount of efforts towards greener ICT (Information and Communications Technology). The proposed approaches, however, have mainly focused on designing networks and their elements so that the power consumed is proportional to the traffic load. In particular, the proposed approaches include the design of new energy-efficient hardware [34], energy efficient routing mechanisms [35] [36], putting devices in sleep mode [11], [37], etc. These approaches address important issues in the core of the network. However, they should be complemented with new techniques to save energy in the end systems (i.e., at the edge of the network) which are responsible for the major share of the Internet power consumption [20] [23]. A comprehensive survey on energy efficient approaches to networking can be found in [38] and a more concentrated survey of approaches in green mobile networks can be found in [39]. In this Thesis, however, we focus mainly on the conventional Internet and any further discussion on mobile networks is beyond the scope of the Thesis.

In this chapter, we discuss related work in energy efficiency in file distribution focusing on P2P and content distribution networking. We also briefly discuss variants of file distribution problem with other optimization goals. The most important one being the minimization of time to finish a file distribution process.

2.1. Energy Efficiency in File Distribution

File distribution is one of the basic tasks in ICT, which is also the focus of study in this Thesis. While many other studies have been conducted on file distribution for optimization of other parameters, specially the distribution time, our focus is on energy consumption during a file distribution process. Next, we classify energy efficient studies based on P2P and content distribution networking.

2.1.1. Focus on P2P Networking

The literature is full of research on energy efficient transfers using P2P. The energy models in these works have mainly considered, proxying [40] [41] [42], sleep-and-wake [43], task allocation optimization at processor level [44] [45], message reduction [46], overlay structure optimization [47], and location-based techniques [48], to reduce energy consumption. We focus on scheduling of the file distribution mechanism such that the hosts minimize the upload and download time they are on for receiving the file [49] [50] [51] [52]. This adds one more model to the above list.

An adaptive algorithm AdaBT is proposed in [53], which dynamically selects the most energy efficient option between legacy bittorrent and a proxy based approach. They argue that for low upload speeds legacy bittorrent is worse than proxy based approaches, opposite if upload speeds are high. Forming groups based on energy consumption is studied in [54]. Essential idea is to favour the group of low energy devices by allocating higher download bandwidths to them because they are operating at lower energy budgets. A mixed integer linear programming model for energy efficient peer selection is developed in [55]. A very important and complementary approach is provided in [56] [57] for greening P2P file transfers taking content pollution in consideration. A two layered model for energy efficient P2P networks is described in [64].

[58] propose a method complementary to proxying and present results by classifying hosts in homogeneous and heterogeneous groups. Their approach is mainly to conduct simulations unlike ours. Similarly, [59] investigates green bittorrent via simulations. Like ours, their approach also tries to push the energy consumption from the server side to the hosts that are actually up for receiving. Nevertheless, their evaluations are confined to simulation and P2P networks.

Energy efficiency in IP/WDM networks are studied in [60]. Energy reduction in P2P networks for discovering frequent item set in unstructured P2P networks forms the basis for research in [61]. The approach of migrating the services provided by a data center to bittorrent based Set-Top-Box is presented in [62] and [63].

[45] [44] discuss a model to show the relation of the amount of computation and the total power consumption of peer to peer systems considering a web type application on P2P overlay networks. They also discuss algorithms for allocating a process to a computer so that the deadline constraint is satisfied and the total power consumption is reduced.

Energy consumption in P2P systems have been the topic of study in various studies. For details about more similar studies, we refer the reader to [47] and [66].

A substantial amount of work is done in the area of mobile devices and wireless devices to address the problem of energy efficient P2P file sharing. However, our work is complementary to these approaches and does not conflict with any of the solutions whose details are provided in the next sections.

2.1.1.1. Mobile P2P

A good amount of research exists in making mobile P2P more energy efficient. In particular, Kelenyi et. al, proposed the problem in [67] and after which they have provided many solutions. Like addressed before, proxying is a good solution for mobile P2P as well [68] [69] [70] [40] [71]. They also propose an energy efficient bittorrent content sharing mechanism for mobiles via cloud services [72]. A demo of energy efficient P2P mobile video streaming demo and benchmarking platform are given in [73] and a detailed study is provided in [74]. Many energy efficient techniques for P2P file sharing in mobile phones are tested in [75] and a cloud approach is proposed as a new solution. Energy efficient P2P mobile communications based on context awareness are provided in [76].

[77]

2.1.1.2. Wireless P2P

Energy efficient broadcasting in wireless ad-hoc and sensor networks has been studied comprehensively. It is so because the devices mostly operate on battery and reduction in energy is required to keep the device functional for a longer period. The main approaches for energy reduction in wireless networks are through optimizing the transmission levels of the relay nodes [78], use of network coding to minimize the number of transmissions required [79], [80]. [81] provides a solution to energy efficient wireless P2P file sharing by proposing a new protocol. All these proposals, however, are not directly applicable outside the wireless regime because of the broadcast nature of the wireless channel. Our approach serves as complementary to these because we assume that no two hosts are present in the same wireless collision domain.

2.1.2. Energy Efficient Content Distribution and Replication

Our work can be extended from P2P file sharing to content distribution networks and content replication as well. So we discuss some of the already present relevant solutions. [82] proposes a heuristic based on an integer linear program with an objective to find a feasible routing so that the energy consumption is minimized. In contrast, we focus on how to minimize the upload and download time of each server. Our approach is also complementary to the solutions provided in Ph.D. thesis [83]. In [84], authors study the local load balancing of servers to shutdown local servers in a cluster. The goal is to shut down the whole cluster using a global load balancing in [85].

Studies related to content distribution are of interest in [86], [87] but their approach and problem definition are altogether different from ours. Their focus is on moving the content the closest to the users through caching so that the data travels lesser distance. Algorithms designed for content replication and dissemination in cloud have to consider energy as a key parameter of optimal operation [88]. Energy efficient replication in cloud computing data centers is studied in [16].

[89]

2.2. Different Studies on File Distribution

Finally, we discuss the research with time minimization as the goal and compare it with energy minimization. An important amount of effort has been dedicated to study the completion time in a file distribution process [90] [91] [92] [33]. The minimization of the average finish time in P2P networks is considered in [46, 93, 94]. Mundinger et al. [95] present a theoretical study to derive the minimum time associated to a P2P file distribution process. However, a scheme guaranteeing file distribution with minimum completion time does not generally lead to minimize the energy consumption and may consume very high energy which may be an order of magnitude higher [31]. Indeed, schemes with the same distribution time may have different energy costs too. It is so because the schemes that minimize time are energy agnostic and may keep a host with high power consumption on for a longer period of time. Similarly, minimizing average finish time also does not minimize energy.

Chapter 3

Energy Efficient File Distribution: Model and Complexity Analysis

One of the biggest challenges in energy efficients methods is to accurately capture the notion of how energy is being spent. In this chapter, we begin with focus on devising an appropriate model for the energy consumed during a file distribution process. The assumptions and problem formulation are stated with results on the computational complexity of the problem. We eventually prove that the problem of minimizing energy consumption during a file distribution process is NP-hard. We also prove some other variants of the problem to be NP-hard and devise a method to relax parameters so that the problem can be solved in polynomial time. In the rest of the Thesis, we work with the version of the problem that are computationally tractable.

3.1. System Model and Assumptions

We consider a system of $n + 1$ hosts ($n \geq 1$) that are fully connected, where every host is able to send messages to every other host. One of these hosts, called the *server* and denoted by S, has initially a file of size B that it has to distribute to all the other hosts, which we call the *clients*. We assume that the file is divided into $\beta \geq 1$ blocks of equal size $s = B/\beta$. The set of hosts is denoted as $\mathcal{H} = \{S, H_0, H_1, ..., H_{n-1}\}$, and the set of blocks as $\mathcal{B} = \{b_0, b_1, ..., b_{\beta-1}\}$. We will also use a set of indexes, defined as $\mathcal{I} = \{S, 0, \dots, n-1\}$. For simplicity of notation and presentation, we will often use an index $i \in \mathcal{I}$ to denote a host, and even talk about host i as H_i (or S when $i = S$). All the hosts are identical with respect to the processing speed and memory. We also require that there are no packet losses in the system, hence, no two hosts can belong to the same collision domain, wired or wireless.

All the hosts in $\mathcal{H}$ can upload blocks of the file to other hosts (initially only S can do so). A client can start uploading block b_i only if it has received b_i completely. Host H_i

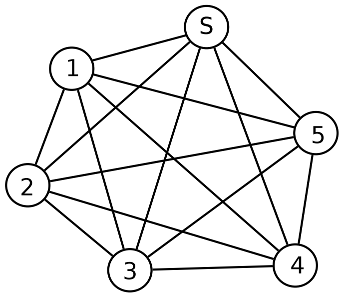


Figure 3.1: Topology of the network. We assume an overlay of all the nodes which is a complete graph.

has upload capacity u_i and download capacity d_i, for $i \in \mathcal{I}$. (Observe that the server has upload capacity u_S.) We assume that all capacities are integral.

We require that time in the file distribution process is slotted. One or more blocks may be transfered in one slot, which is the time taken by a host to upload the block(s) to the receiver. Note that, in general, slot duration may vary from one slot to the next because the upload capacities may be different but the block size remains the same. This formulation of the problem is NP-complete as shown in Theorem 1.

In this Thesis, we consider only the energy consumed by hosts during the file distribution process. We do not consider the energy consumed by other network devices. Therefore, in our model, energy consumption has the following two components:7

1. Each host $i \in \mathcal{I}$, just for being on, consumes power P_i (when a host is off, we assume that it consumes no power).

2. A host consumes energy while being switched on or off. If host $i \in \mathcal{I}$ takes time α_i to switch on or off, the energy consumed by switching is $P_i \cdot \alpha_i$. Usually, this on/off time α_i is less than or equal to two seconds [96, 97]. Unless otherwise stated, we assume $\alpha_i = 0, \ \forall i \in \mathcal{I}$, i.e., switching a system on or off is instantaneous. Hence, energy consumed during switching on/off a host is 0.

A host is said to be *active* in a time slot if it is receiving or serving blocks in the slot. Otherwise, it is said to be *idle*. We assume that $\forall i \in \mathcal{I}$ power consumption P_i is constant, irrespective of whether it is using its upload capacity, download capacity, both, or none. Then, energy Δ_i consumed by an active host $i \in \mathcal{I}$ in one slot can be computed as follows.

$$\Delta_i = \frac{P_i s}{u} = \frac{P_i B}{u\beta} \tag{3.1}$$

Without loss of generality, we assume that $\Delta_0 \leq \cdots \leq \Delta_{n-1} \leq \Delta_S$. We also assume that there are no failures in transmission, i.e., each block is transfered exactly once to a host and that there is no propagation delay.

Table 3.1: Some of the notation used in this work.

Symbol	Definition
n	Total number of clients
H_i	i^{th} client
S	Server, host that has the file initially
$\mathcal{I}$	Set of host indexes
β	Number of blocks into which the file is divided
b_j	j^{th} block
B	Size of the file in bits
s	Size of a block in bits
u	Upload link speed (bits/s)
d	Download link speed (bits/s)
k	Ratio of the download to upload capacity (d/u)
P_i	Power consumed by host i when on (in Watt)
Δ_i	Energy consumed by host i involved in a block transfer in a slot
τ	Any arbitrary time slot
z	A scheme to accomplish the distribution process
$c^z_{j,i}$	Energy to transfer block b_j to host H_i under z
$serv(j,i)$	Index of the host that serves b_j to host H_i
$\mathcal{I}^z_\tau$	Set of active hosts in slot τ under scheme z

3.2. Problem Formulation and its Complexity

We define a *file distribution scheme*, or *scheme* for short, as a schedule of block transfers between hosts such that, after all the transfers, all the hosts have the whole file. Observe that a scheme must respect the model previously defined. Then the problem we study is defined as follows.

Definition 1. *(EOFD Problem) The* Energy Optimal File Distribution Problem *is the problem of finding or designing a file distribution scheme that minimizes the total energy consumed.*

We show that the problem is NP-Complete, even if switching on and off consumes no energy (i.e., $\alpha_i = 0, \forall i \in \mathcal{I}$). The following theorem establishes this fact.

Theorem 1. *Assume that time is slotted, that a host can upload and download at the same time slot, and that a host can upload to more than one host in the same slot. The problem of minimizing the energy consumption of file distribution is NP-hard if hosts can have different upload capacities and power consumptions, even if the energy consumed to switch on/off is zero (i.e.,* $\alpha_i = 0,\ \forall i \in \mathcal{I}$*).*

Proof: We show a reduction from the following NP-Complete problem (see [98]):
Partition Problem:

Input: A set of integers $A = \{a_1, a_2, \cdots, a_n\}$, $0 < a_i < M$ for every i, $\sum_{j=1}^{n} a_j = 2M$.
Question: Is there a subset $\{a_{j_1}, a_{j_2}, \cdots, a_{j_K}\} \subseteq A$ such that $\sum_{t=1}^{K} a_{j_t} = M$?

We are given an instance I of the Partition Problem, that is, $A = \{a_1, a_2, \cdots, a_n\}$, $0 < a_i < M$ for all i, $\sum_{i=1}^{n} a_i = 2M$. We define an instance $\widehat{I}$ of our problem, as follows. The set of hosts is $\{S, R\} \cup N$, where $N = \{1, 2, \cdots, n\}$. S is the server who initially holds the file of $2M$ blocks of size 1. The upload capacities are $(2n+1)M$ for S, a_i for every $i \in N$, and 0 for R. The download capacities are 0 for S, $2M$ for every $i \in N$, and M for R. The power consumptions are $E_S = M + 1$, $E_i = a_i$ for every $i \in N$, and $E_R = 4M + 2$. The bound for the total energy is $E = 12M + 5$. We have to show that there is a solution to I iff there is a solution to $\widehat{I}$.

Assume there is a solution to I, that is, a subset $\{a_{j_1}, a_{j_2}, \cdots, a_{j_K}\} \subseteq A$ such that $\sum_{t=1}^{K} a_{j_t} = M$. We describe a solution for $\widehat{I}$. First the source S will send the $2M$ blocks to each of the hosts in N, and M blocks to R, in one time slot (note that its upload capacity is $(2n+1)M$). This will use $E_S + E_R + \sum_{i=1}^{n} E_i = 5M + 3 + \sum_{i=1}^{n} a_i = 7M + 3$ energy. Hosts $j_1, j_2, \cdots, j_K$, whose total upload capacity is $\sum_{t=1}^{K} a_{j_t} = M$, will then send in one time slot the rest M blocks to R. This will use $E_R + \sum_{t=1}^{K} E_{j_t} = 4M + 2 + \sum_{t=1}^{K} a_{j_t} = 5M + 2$ energy. Thus, the total energy used will be $12M + 5 = E$. We have thus established a solution to $\widehat{I}$, which uses no more than E energy.

Assume there is a solution for $\widehat{I}$, that uses no more than $E = 12M + 5$ energy. As R needs to download $2M$ blocks, and can download at most M blocks in one time slot, it must be active in at least two time slots. If it will be active in more time slots, then the total energy consumed will be at least $3E_R$; this is a contradiction, since $3E_R > E$. Thus, R must be active in exactly two time slots, and in each of them it must receive exactly M blocks.

The energy used by R in these two time slots is $2E_R$. Also, there is at least one round in which S uploads, and thus uses E_S energy. Last, for each host in $i \in N$, in the first time slot when it downloads blocks it uses E_i energy and does not upload any block. The total energy used is at least $2E_R + E_S + \sum_{i=1}^{n} E_i = 11M + 5$. Thus, a total of $11M + 5$ is used in which R can download at most M blocks. So at most M energy can be used by hosts who upload the other M blocks to R. This can be done only by hosts in N. But the total energy to be used is at least M (since a host $i \in N$ who uploads at most a_i blocks uses $E_i = a_i$ energy). We conclude that R downloads M blocks from hosts in N whose total energy is M. Thus, if in one of these time slots, R downloads M blocks from the K hosts $\{j_1, j_2, \cdots, j_K\} \subseteq N$, this means that $\sum_{t=1}^{K} a_{j_t} = M$. But then the set of K integers $\{a_{j_1}, a_{j_2}, \cdots, a_{j_K}\}$ is a solution to the instance I of the Partition Problem.

■

While EOFD is shown to be NP-complete. Hence, we want to make the problem tractable by relaxing some of the constraints of the original problem. However, we must find parameters to relax so that the problem remains useful as well as solvable in polynomial time. In Theorem 1 it was allowed for a host to multiple hosts and upload at any capacity up to its maximum upload capacity. We consider relaxing these conditions, i.e., we restrict that a host can upload only at its full capacity (Theorem 2), and in another one we restrict that a host can upload to at most one host in a slot (Theorem 3). We show that the problem remains NP-complete with these different assumptions too.

3.2.1. Other variants of the problem

We first show that when the hosts are restricted to send in full speed the problem is NP-complete (Theorem 2). We then show that even without this assumption (of sending in full speed) the problem is still NP-complete (Theorem 3). In our reduction, we use a special form of the partition problem ($P3$).

Partition-1 ($P1$) *Input:* A set of integers $A = \{a_1, a_2, \cdots, a_n\}$, $\sum_{j=1}^{n} a_j = 2M$.

Question: Is there a subset $\{a_{j_1}, a_{j_2}, \cdots, a_{j_K}\} \subset A$ such that $\sum_{t=1}^{K} a_{j_t} = M$?

Problem $P1$ is known to be NP-Complete (see [98]). From $P1$ it is easy to show that also the following problem $P2$ is NP-Complete:

Partition-2 ($P2$)

Input: A set of integers $A = \{a_1, a_2, \cdots, a_{2n}\}$, $a_i > 0$, $\sum_{j=1}^{2n} a_j = 2M$.

Question: Is there a subset $\{a'_1, a'_2, \cdots, a'_n\} \subset A$ such that $\sum_{j=1}^{n} a'_j = M$?

Using $P2$ it is easy to show that the following Problem $P3$ is also NP-Complete:

Partition-3 ($P3$)

Input: A set of integers $A = \{x_1, x_2, \cdots, x_{2n}\}$, $M > 0$, $M < x_i < 2M$ for all i, $\sum_{j=1}^{2n} x_j = 2(n+1)M$.

Question: Is there a subset $\{x'_1, x'_2, \cdots, x'_n\} \subset A$ such that $\sum_{1}^{n} x'_j = (n+1)M$?

(To reduce $P2$ to $P3$, given an instance of $P2$, define an instance of $P3$ by $x_i = a_i + M$.)

Theorem 2. *Assume that time is slotted, that hosts must upload at their full capacity, and that no host can upload to more than one host in the same slot. The problem of minimizing the energy of file distribution is NP-hard if hosts can have different upload capacities and power consumptions, even if* $\alpha_i = 0, \forall i \in \mathcal{I}$.

Proof: We show a reduction from Problem $P1$. We are given an instance I of $P1$

$$A = \{a_1, a_2, \cdots, a_n\}, a_i \text{ an integer for } i = 1, 2, \cdots, n, \quad \sum_{j=1}^{n} a_j = 2M. \tag{3.2}$$

The set of users is $\{S, R\} \cup N$, where $N = \{1, 2, \cdots, n\}$. S is the user who initially holds the file of $2M$ blocks of size 1. The upload capacities are $2M$ for S, a_i for every $i \in N$, and 0 for R. The download capacities are 0 for S, $2M$ for every $i \in N$, and M for R. The power consumptions are $E_S = 1$ for S, $E_i = a_i$ for every $i \in N$, and , $E_R = 2n + 4M + 1$ for R. The bound for the total energy is $E = 5n + 12M + 2$.

We have to show that there is a solution to I iff there is a solution to $\widehat{I}$. Assume there is a solution to I, i.e., there exists

$$\{a_{j_1}, a_{j_2}, \cdots, a_{j_K}\} \subset A, \quad \sum_{t=1}^{K} a_{j_t} = M. \tag{3.3}$$

Assume there is such a solution for I. We describe a solution for $\widehat{I}$. First S will send all the $2M$ blocks to all users in N in n rounds. This will use $\sum_{i=1}^{n}(E_S + E_i) = nE_S + \sum_{i=1}^{n} a_i = n + 2M$ energy (by (3.2)). Users $j_1, j_2, \cdots, j_K$, can then send in one round (by (3.3)) M of the blocks to R (whose download speed is M), and in the next round the rest M blocks. This will use $2(E_R + \sum_{t=1}^{K} E_{j_t}) = 2(E_R + \sum_{t=1}^{K} a_{j_t}) = 2E_R + 2M = 4n + 10M + 2$ energy. Thus, the total energy used will be $(n+2M)+(4n+10M+2) = 5n+12M+2 = E$. We have thus established a solution to $\widehat{I}$, which uses no more than E energy.

Assume there is a solution for $\widehat{I}$, that uses no more than E energy. Since the download capacity of R is M, and S must send at full speed, it follows that R must receive the blocks only from the users in N. As R needs to get $2M$ blocks, and can get at most M blocks in one round, it must be active in at least two rounds. If it will be active in at least 3 rounds , then the total energy consumed will be at least $3E_R > E$, a contradiction. Thus R must be active in exactly two rounds, and in each of them it must receive exactly M blocks. Assume R receives the M blocks in the first round from K users in $\{j_1, j_2, \cdots, j_K\} \subset N$. Since these users must have transmitted in full speed, this means that $\sum_{t=1}^{K} a_{j_t} = M$. This means that there a solution to the instance I. ■

We now show that problem is still hard even when we do not require the hosts to receive in full speed:

Theorem 3. *Assume that time is slotted, that hosts do not have to upload at their full capacity, and that no host can upload to more than one host in the same slot. The problem of minimizing the energy of file distribution is NP-hard if hosts can have different upload capacities and power consumptions, even if $\alpha_i = 0, \forall i \in \mathcal{I}$.*

Proof: We show a reduction from Problem $P3$. We are given an instance I of $P3$,

$$A = \{x_1, x_2, \cdots, x_{2n}\}, M > 0, \forall i\; M < x_i < 2M, F = (n+1)M, \quad \sum_{j=1}^{2n} x_j = 2F. \tag{3.4}$$

We define an instance $\widehat{I}$ of our problem. The set of users is $\{s, r\} \cup N$, where $N = \{1, 2, \cdots, 2n\}$. S is the user who initially holds $2(n+1)M$ blocks os size 1. The upload capacities are $2F$ for S, x_i for every $i \in N$, and 0 for R. The download capacities are 0 for S, $2F$ for every $i \in N$, and F for R. The power consumptions are $E_S = 1$ for S, $E_i = \frac{x_i}{F}$ for every $i \in N$, and $E_R = 2n + 4.5$ for R. The total energy bound is $E = 6n + 13$. We have to show that there is a solution to I iff there is a solution to $\widehat{I}$.

Assume there is a solution to I, i.e.,

$$\{x_{j_1}, x_{j_2}, \cdots, x_{j_n}\} \subset A \quad , \quad \sum_{t=1}^{n} x_{j_t} = F. \tag{3.5}$$

S will send all the blocks to every $i \in N$ in $2n$ rounds. This will use a total of $2nE_S + \sum_{i=1}^{2n} E_i = 2n + \sum_{i=1}^{2n} \frac{x_i}{F} = 2n + 2$ energy (by (3.4)). Users $x_{j_1}, x_{j_2}, \cdots, x_{j_n}$, whose sum of upload capacities is $\sum_{t=1}^{n} x_{j_t} = F$ (by (3.5)), will then send in one round F blocks to R , and in the next round the rest F blocks. This will use $2(E_R + \sum_{t=1}^{n} E_{j_t}) = 2(E_R + \sum_{t=1}^{n} \frac{x_{j_t}}{F}) = 4n + 11$ energy. Thus, the total energy used will be $(2n+2) + (4n+11) = E$.

Assume there is a solution to $\widehat{I}$, that uses no more than E energy. In particular, we can assume that this solution uses minimum energy. Due to its downloading speed, R must be active in at least two rounds. If it is active in more than two rounds, then the amount of energy used is at least $3E_R > E$, a contradiction. So, R is active in exactly two rounds, and in each of them it must receive exactly F blocks.

If in any of these rounds R receives blocks from S, the energy consumed in this round will be $E_S + E_R = 2n + 5.5$. Alternatively, since a user cannot send and receive in the same round, we can delay all messages sent to R until all users in N got all blocks from S (they cannot get from R, since its upload capacity is 0). R can get $F = (n+1)M$ blocks from any $n+1$ users in N (since user i can send $x_i > M$ blocks in one round). We can take the $n+1$ users which correspond to the $n+1$ smallest x_i's, whose sum is clearly bounded by F. Since user i uses $\frac{x_i}{F}$ energy, when we sum up the energy used by these $n+1$ users we conclude that the energy used in this round will be bounded by $E_R + \frac{F}{F} = 2n + 5.5$. In other words, we can assume that R received the information from the users in N.

We turn now to the users in N.

1. If user i gets the blocks only from S, then the energy consumed is $= \; E_S + E_i = 1 + \frac{x_i}{F}$.

2. If user i gets the blocks from S and from some other users in N, then the energy consumed is larger than $E_S + E_i = 1 + \frac{x_i}{F}$.

3. If user i gets the $2F$ blocks only from users in N, then, since user j uploads at most x_j blocks at the cost of $\frac{x_j}{F}$ energy, user i will receive the $2F$ blocks with energy of at least $\frac{2F}{F} + \frac{x_i}{F} = 2 + \frac{x_i}{F}$.

Thus, the energy used in order for user $i \in N$ to get all blocks is at least $1 + \frac{x_i}{F}$. So the energy needed for all users in N is at least $\sum_1^{2n}(1 + \frac{x_i}{F}) = 2n + \sum_1^{2n} \frac{x_i}{F} = 2n + 2$.

This means that the amount of energy that can be spent in order to deliver the blocks to R (in two rounds) is bounded by $4n + 11$ (since the total amount of energy is bounded by $6n + 13$. $4n + 9$ energy will be used by R . Therefore the total amount of energy to be used by the users in N is bounded by 2. But since sending F blocks uses at least $\frac{F}{F} = 1$ energy, this means that in each such round the users in N must deliver exactly F blocks. Assume R receives the F blocks in the first round from K users in $\{j_1, j_2, \cdots, j_K\} \subset N$. This implies that $\sum_{t=1}^{K} x_{j_t} = F$. Denote $y_i = x_i - M$. Then by (3.4) we get

$$0 < y_i < M \ , \ \sum_{t=1}^{2n} y_t = 2M. \tag{3.6}$$

If $K > n$, then by (3.6) $\sum_{t=1}^{K} x_{j_t} = \sum_{t=1}^{K}(M + y_{j_t}) = kM + \sum_{t=1}^{K} y_{j_t} > kM \geq F$, a contradiction . If $K < n$, then by (3.6) $\sum_{t=1}^{K} x_{j_t} = \sum_{t=1}^{K}(M + y_{j_t}) = kM + \sum_{t=1}^{K} y_{j_t} < (n-1)M + \sum_{t=1}^{2n} y_t = (n-1)M + 2M = F$, a contradiction. This means that $K = n$, and therefore $x_{j_1}, x_{j_2}, \cdots, x_{j_n}$ is a solution to I, as required. ■

3.3. Towards solving the problem in polynomial time

The last section proves three variants of the problem as NP-complete. To get better insight on on the complexity of the problem, in this section, we formulate the problem as a search in a directed acyclic graph, called *State Transition Graph, abbreviated as STG* henceforth, as shown in Fig. 3.2. For this purpose, we define notion of a state ψ, which is a $n \times \beta$ matrix representing the nodes in STG.

The elements a_{ij} of a state ψ take values as follows:

$$a_{ij} = \begin{cases} \text{Index of the host who served block } j \text{ to } H_i \\ 0 \quad \text{If block } j \text{ is not yet downloaded by } H_i \end{cases}$$

$$\psi = \begin{pmatrix} S & S & \cdots & 0 \\ S & H_0 & \cdots & S \\ \vdots & \vdots & \ddots & \vdots \\ H_1 & H_1 & \cdots & S \end{pmatrix}_{n \times \beta} \tag{3.7}$$

Clearly, $a_{ij} \in \{S, H_0, H_1, \cdots, H_{n-1}\}$. Thus, any state ψ can be represented as shown in Eq. 3.7. A state ψ is defined a $n \times \beta$ matrix. n rows for the clients who download the file and β columns corresponding to each block of the file. Note that the initial state is the one in which all the elements are 0. On the other hand, final state has all non-zero entries. If there are l non-zero entries, then the state belongs to level l in Fig. 3.2. A directed edge $\psi \to \psi'$ in STG signifies that given a state ψ, at the start of a slot τ, blocks can be transfered so that state ψ' is achieved. The weight of the edge is equal to the

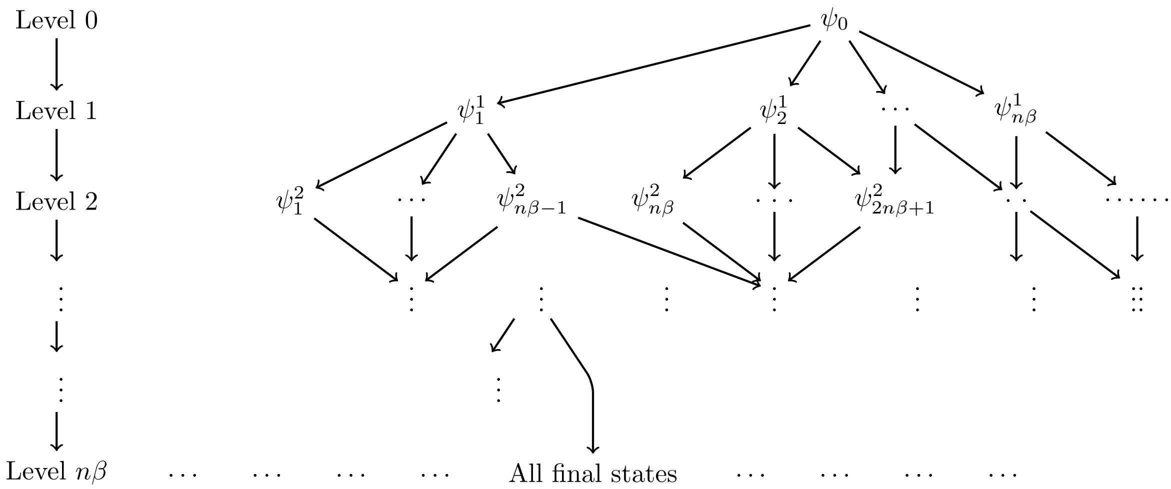


Figure 3.2: *State Transition Graph*: Representation of all the possible schemes in the form of a directed acyclic graph.

energy consumed in the state transition. Let the set of hosts involved in block transfers during slot τ is given by $\mathcal{I}_\tau$ (which is constant throughout the slot), then the energy consumed during τ is given by

$$E_\tau = \sum_{i \in \mathcal{I}_\tau} P_i \cdot \text{duration of } \tau \tag{3.8}$$

STG contains all the possible states during any file transfer scheme. Any scheme z can be represented as a path from the initial state to one of the final states at level $n\beta$ in STG. It captures all the possible schemes from the initial state to final states. It can be easily seen that the number of states in STG is exponential in n and β. To find an energy optimal scheme we have to find the shortest path from initial to final. However, the explosion of states make it a hard problem, as we have already seen in Section 3.2.

So our goal is to look for cases under which the search for optimal state(s) can be performed efficiently. To be able to find energy optimal path to a final state, we should be able to prune the STG efficiently. STG as in Fig. 3.2 does not have much structure that can be exploited to efficiently search a path to optimal final state. However, if we enforce each slot to have the same duration, then STG as shown in Fig. 3.3 becomes far more structured. In this case, the weight of the edges between states follow triangle inequality. Thus, an edge from one state to the other is also the shortest path between them.

We observe the following two properties for STG in Fig. 3.3:

1. *Isomorphism:* Two states at the same level are isomorphic *iff* they can be transformed to each other via column exchanges.

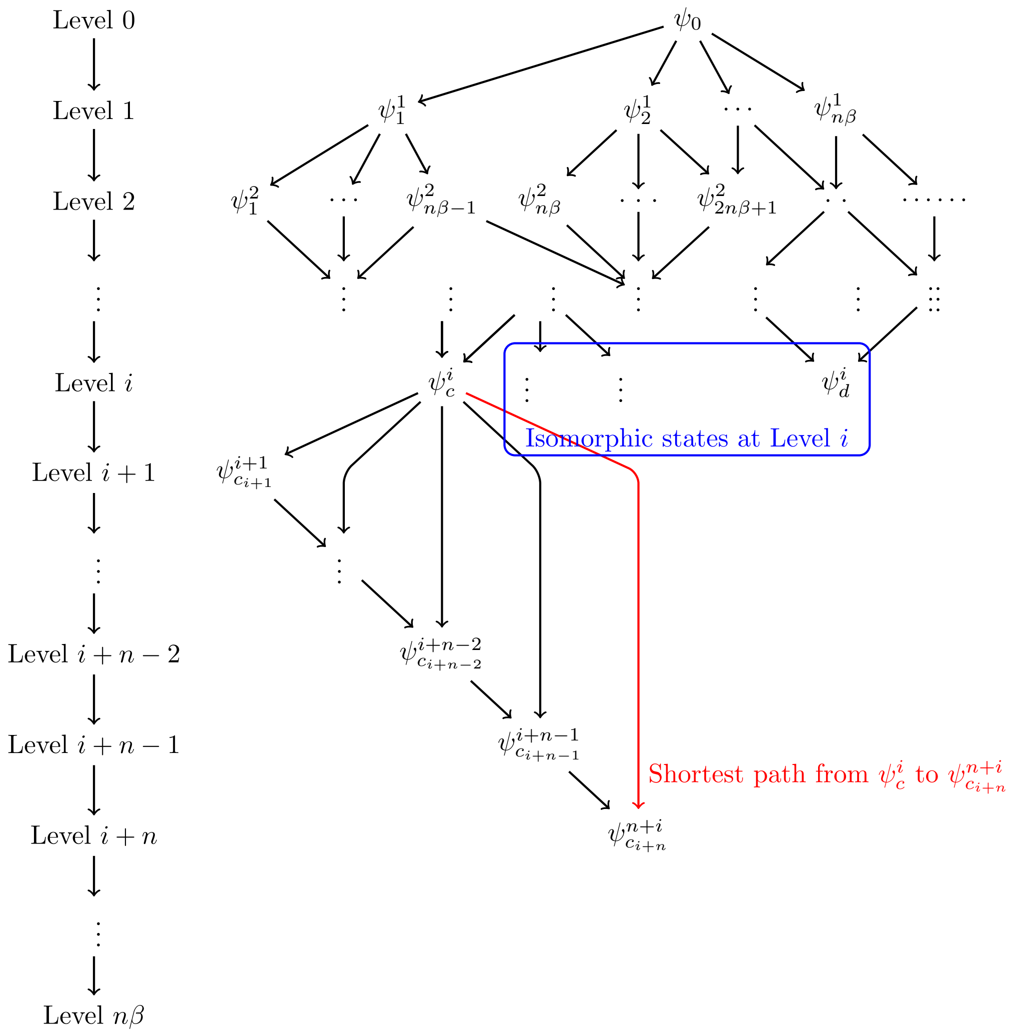


Figure 3.3: STG when each host i has upload capacity u and download capacity d.

We provide an intuition of this property through an example. The two states shown below are isomorphic.

$$\psi_1 = \begin{pmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & S \end{pmatrix}, \quad \psi_2 = \begin{pmatrix} 0 & S & 0 \\ 0 & 0 & S \\ S & 0 & 0 \end{pmatrix}$$

They both are at level 3, and consume same energy, i.e., $(3P_S + P_1 + P_2 + P_3) \cdot$ (slot duration). To transform ψ_1 to ψ_2, first exchange column 2 and 3 of ψ_1, then exchange column 1 and 2 of the thus obtained state. Similarly, both the above states are isomorphic to the following state.

$$\psi_3 = \begin{pmatrix} 0 & 0 & S \\ 0 & S & 0 \\ S & 0 & 0 \end{pmatrix}$$

2. *Trineq:* Triangle inequality holds for the weight of edges, i.e., for any three states ψ_1, ψ_2 and ψ_3 with direct edges and cost to reach as shown in Fig. 3.4

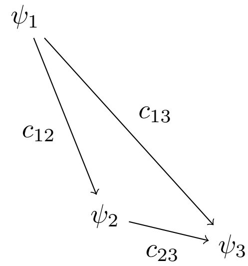


Figure 3.4: Triangle inequality holds for STG in Fig. 3.3, i.e., $c_{13} \leq c_{12} + c_{23}$.

With these two properties, since there are many more states which can be optimal, it is relatively easy to prune the STG in Fig. 3.3. The next section provides necessary definitions and conditions for solving EEFD optimally.

3.4. Solving EEFD Optimally

As discussed in the above section, in order to make the problem computationally tractable, we add some constraints. We require that the file is divided in blocks of equal size s. Padding may be used if the last block has size less than s. We state the most important constraint, from now on, throughout the Thesis, unless stated otherwise, we assume that $u_i = u, \forall i \in \{S, 0, 1, \cdots, n-1\}$. This simplifies the problem because it implies that all time slots must be of the same duration as long as $u \leq d_i \; \forall i \in \{S, 0, 1, \cdots, n-1\}$. For similar reasons, we also assume that $d_i = d, \forall i \in \{S, 0, 1, \cdots, n-1\}$.

We now make the definition of a slot more precise. We define a slot as the time duration in which exactly one block is transfered and is given as,

$$\tau_{duration} = \frac{s}{\min\{u, d\}} \tag{3.9}$$

Hence, energy consumed by host i per slot is given by

$$\Delta_i = P_i \cdot \frac{s}{\min\{u, d\}} \tag{3.10}$$

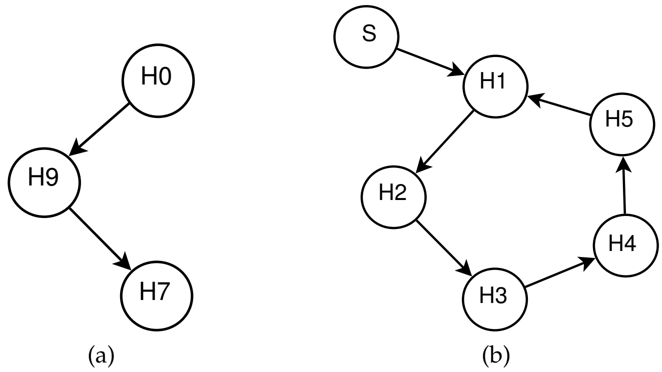


Figure 3.5: Two transfer graphs are shown in which each connected component has mutually disjoint hosts. Since energy consumed in switching on/off is zero, these two transfer graphs can be done one after the other without any impact on energy consumption of the scheme.

We further assume that both u and d are related by a positive integer $k \geq 1$, such that, $k = \frac{u}{d}$ or $k = \frac{d}{u}$, depending on whether $u \geq d$ or $d > u$. Then, if the process of file distribution starts at time $t = 0$, the first slot spans time $[0, 1)$, the second slot $[1, 2)$, and so on. In each slot, a host can download from a set of hosts, or it can upload to some hosts, as the case may be. Some of the notations used in the work are provided in Table 3.1. We assume for simplicity that the upload or download of the blocks start at the beginning of a slot.

3.4.1. Normal Schemes

To rule out redundant and uninteresting schemes, we will consider only what we call *normal schemes*. Observe that the block transfers of a scheme z in a slot τ can be modeled as a *block transfer graph* which is a directed graph with the hosts as vertices and block transfers as edges. Then, a *normal scheme* is a distribution scheme in which:

1. There is no idle host, i.e., no host is powered on unless uploading and/or downloading blocks.

2. There is no idle slot, i.e., there are no slots without active hosts.

3. There are no errors and each block can be received correctly in one transmission.

4. There is exactly one connected component of the transfer graph per slot (Fig. 3.5).

Since we assume that energy consumed in switching on/off is zero, transfer between two or more mutually disjoint group of hosts can be shifted to different slots, with each slot having only one connected component (Fig. 3.5).

We denote the set of normal schemes with parameters n, β, and k by $\hat{\mathcal{Z}}_k^{n,\beta}$. From now onwards, throughout the Thesis, by schemes we mean normal schemes. It is easy to observe that any optimal scheme can be transformed into a normal scheme that is also optimal. The time taken to finish the distribution process may increase, though, as two transfer graphs with mutually exclusive hosts cannot exist in a slot because of this restriction. Fortunately, as we will see later, proposed optimal schemes are not affected by this restriction.

Let us consider parameters n, k, and β of the file distribution energy minimization problem. Let us define the set of all possible schemes with these parameters by $\mathcal{Z}_k^{n,\beta}$. Let $E(z)$ be the energy consumed by scheme $z \in \mathcal{Z}_k^{n,\beta}$.

Definition 2. *Among all normal schemes, a scheme $z_0 \in \mathcal{Z}_k^{n,\beta}$ is* energy optimal *(or optimal for short) if $E(z_0) \leq E(z), \forall z \in \mathcal{Z}_k^{n,\beta}$.*

Our objective in this Thesis is to find optimal schemes.

3.5. Energy Costs

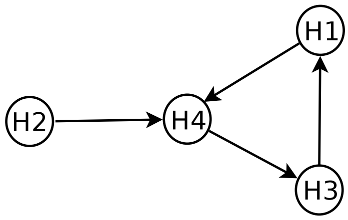
We have already discussed about the nodes in a block transfer graph and the state transition graph. Now we turn our attention towards the weight of edges. Let us consider scheme $z \in \hat{\mathcal{Z}}_k^{n,\beta}$. Denote with $\mathcal{I}_\tau^z \subseteq \mathcal{I}$ the indexes of the set of active hosts in time slot τ under scheme z. Let τ_f^z be the final slot of scheme z, i.e., the time slot of z in which the distribution of the file is completed. In what follows, we compute the cost of a slot (forms edge weight for State Transition Graph (STG)) and cost of a block (forms edge weight for Block Transfer Graph (BTG)).

3.5.1. Cost of a Slot

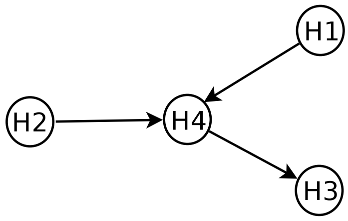
The cost of a slot defines edge weight for STG.

Definition 3. *The* cost of slot τ *under scheme z, denoted c_τ^z, is the energy consumed by all active hosts $\mathcal{I}_\tau^z$ in τ, i.e, $c_\tau^z = \sum_{i \in \mathcal{I}_\tau^z} \Delta_i$, and the energy consumed by the scheme z is $E(z) = \sum_{\tau=1}^{\tau_f^z} c_\tau^z = \sum_{\tau=1}^{\tau_f^z} \sum_{i \in \mathcal{I}_\tau^z} \Delta_i$.*

The cost of a slot, as defined above, considers the set of hosts that are active in a particular time slot, but it does not take into account which host is serving which block to which host. Any transfer graph with the same set of nodes but different edge connectivity are equivalent with respect to the energy consumption. The number of blocks served in 3.6a is one more than the number of blocks served in 3.6b, with the same energy consumption.



(a) Slot with cycle



(b) Tree slot

Figure 3.6: Two slots with the same power consumption but different set of blocks being transfered.

3.5.2. Cost of a Block

Cost of a block defines edge weight for BTG. For a better insight on energy consumed by schemes, we also associate a cost to a block transfer. The cost of block transfers will be used in the proofs of lower bounds. Let us denote the set of blocks downloaded by host $i \in \mathcal{I}$ in slot τ under scheme z by $\mathcal{S}^z_{i,\tau}$ and the index of the host serving $b_j \in \mathcal{S}^z_{i,\tau}$ as $serv(j,i)$.

Definition 4. *Denote the set of blocks downloaded by host $i \in \mathcal{I}$ in slot τ under scheme z by $\mathcal{S}^z_{i,\tau}$ and the index of the host uploading block $j \in \mathcal{S}^z_{i,\tau}$ as $serv(j,i)$. Also the set of blocks uploaded by $serv(j,i)$ in the same slot τ be $\mathcal{W}^z_{serv(j,i),\tau}$. We define the* cost $c^z_{j,i}$ of a block b_j *received by H_i in slot τ under scheme z as,*

$$c^z_{j,i} = \frac{s}{\min\{u,d\}} \cdot (\mathcal{D}^z_{j,i} \cdot P_i + \mathcal{U}^z_{j,i} \cdot P_{serv(j,i)}) \tag{3.11}$$

where,

$$\mathcal{D}^z_{j,i} = \frac{1}{|\mathcal{S}^z_{i,\tau}|}$$

$$\mathcal{U}^z_{j,i} = \begin{cases} \frac{1}{|\mathcal{W}^z_{serv(j,i),\tau}|} & \textit{if } \mathcal{S}^z_{serv(j,i),\tau} = \emptyset \\ 0 & \textit{Otherwise} \end{cases}$$

An example of a transfer graph to demonstrate above definition is given in Fig. 3.7. As we will see later, optimal schemes have much more symmetrical transfer graphs.

3.5.3. Cost of a Scheme

With the above definition, the sum of the costs of all blocks transferred in slot τ of scheme z should be equal to the cost of the slot τ, c^z_τ. The next result establishes that this is indeed true for all the schemes.

It can be easily seen that the sum of the costs of all the blocks transferred during slot

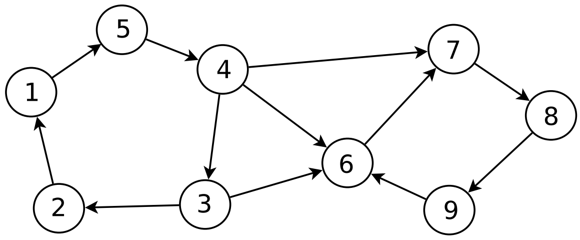


Figure 3.7: Transfer graph with two cycles. Note that only 9 hosts are on but 12 blocks have been transfered. For host 6, $|\mathcal{S}^z_{6,\tau}| = 3$ as three blocks are being received. For host 3, $|\mathcal{W}^z_{serv(j,6),\tau}| = 2$ as it is uploading two blocks. Also for 3, $\mathcal{S}^z_{serv(j,6),\tau} \neq \emptyset$ as it is receiving a block from host 4. Hence, the cost of block that host 3 is uploading to 6 is given by $\frac{1}{3} \cdot P_6 + 0 \cdot P_3$. Similarly, after computing the costs of all the blocks it can be seen that their sum is $\sum_i^9 P_i$.

τ of scheme z is equal to the cost of that slot, i.e.,

$$\sum_{i \in \mathcal{I}^z_\tau} \sum_{b_j \in \mathcal{S}^z_{i,\tau}} c^z_{j,i} = c^z_\tau \tag{3.12}$$

Hence, the energy consumed by the scheme is

$$E(z) = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} c^z_{j,i} = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} \left(\Delta_i \cdot \mathcal{D}^z_{j,i} + \Delta_{serv(j,i)} \cdot \mathcal{U}^z_{j,i} \right) \tag{3.13}$$

The cost of a scheme is same as weight of path from initial to final state in STG. In fact, each path from initial to final state in STG represents a scheme. As mentioned before, our goal is to find the optimal schemes for different cases we have presented. Chapter 4, 5 and 6 address energy efficient schemes and their properties for the cases $d = u$, $u = kd$ and $d = ku$ respectively.

Chapter 4

Optimal Solutions: Download = Upload Capacity

We start presenting the results obtained during the course of the Thesis with a simple case, that is easy to understand and is also a representative of the main idea behind the solutions proposed in the Thesis. Throughout this chapter, we will assume that the set of n clients as well the server S have equal download capacities, i.e, $d_S = d_i = d \; \forall i \in \{0, 1, 2, \cdots, n-1\}$ and equal upload capacities, i.e., $u_S = u_i = u \; \forall i \in \{0, 1, 2, \cdots, n-1\}$. Furthermore, we require that the download capacity is equal to the upload capacity, i.e., $d = u$. So the parameter $k = \frac{d}{u} = 1$. Note that this ensures that all the slots have equal lengths and time taken to upload a block to a host is the same as time taken to download a block in a slot. Let $d_{\max}$ be the maximum of all download capacities $\{d_0, d_1, \cdots, d_{n-1}\}$ and $u_{\min}$ be the minimum of all upload capacities $\{u_S, u_0, u_1, \cdots, u_{n-1}\}$. Then the following lemma helps in proving lower bounds on energy consumption.

Lemma 1. *If $d_{\max} \leq u_{\min}$, then, for any host, restricting download from at most one host in one time slot does not increase the total energy consumption.*

Proof: Let us assume that host i with download capacity d_i is downloading from $l > 1$ hosts. We show that receving from individual hosts at full capacity consumes energy no more than receiving from hosts in parallel, for any i. Let the size of each block be s, since H_i is downloading from l hosts, it is receving l blocks. All the hosts have to be active for at least l slots. Thus, the energy consumed in complete transfer is

$$E_{parallel} = P_i \frac{ls}{d_i} + \left(\sum_{j=1, j \neq i}^{l} P_j \right) \frac{ls}{d_i} \tag{4.1}$$

In case H_i downloads l blocks from hosts sequentially one by one, energy consumed is

$$E_{sequential} = P_i \frac{ls}{d_i} + \left(\sum_{j=1, j \neq i}^{l} P_j \right) \frac{s}{d_i} < E_{parallel}, \quad \forall i \in \mathcal{L} \tag{4.2}$$

■

Since download and upload capacities are equal, from Lemma 1, we can conclude that the best strategy is to download at most one block from at most one host during a slot.

The results presented in this chapter are divided into the following sections. In Section 4.1, we derive lower bounds on the energy consumed by any scheme, and subsequently design optimal schemes in Section 4.2. We analyze the fairness of algorithms with respect to power consumption by individual hosts in Section 4.3. Finally, under an extended model, in Section 4.4 we find the optimal number of blocks in which a file should be divided so as to minimize the energy of optimal schemes.

4.1. Lower Bound

The following theorem provides a lower bound on the energy consumed by any distribution scheme when $k = 1$. The key observation behind this result is that each host has to be active for at least β slots to receive the file, whereas the server has to be active for at least β slots to upload one copy of each block to the clients.

Theorem 4. *The energy required by any scheme z to distribute a file divided into β blocks among n clients when $k = \frac{d}{u} = 1$, satisfies $E(z) \geq \beta \left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i \right) + \max\{0, n - \beta\} \min\{\Delta_S, \Delta_0\}$.*

Proof: The claim to be shown is that if $k = \frac{d}{u} = 1$, then any scheme z consumes energy $E(z)$, given by,

$$E(z) \geq \beta \left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i \right) + \max\{0, n - \beta\} \min\{\Delta_S, \Delta_0\} \tag{4.3}$$

Before proving the claim, we need some supporting arguments.

Lemma 2. *For every block b_j and every client H_i it holds that $\mathcal{D}^z_{j,i} = 1$.*

Proof: Since $d = u$, each host can receive only one block in a time slot. Hence, if block b_j is transferred to client H_i in slot τ, we have $|\mathcal{S}^z_{i,\tau}| = 1$. Then, by definition, $\mathcal{D}^z_{j,i} = 1$. ■

Lemma 3. *For every block b_j served by S to client H_i, it holds $\mathcal{U}^z_{j,i} = 1$.*

Proof: Let S be serving b_j to H_i in slot τ. Then, $\mathcal{S}^z_{S,\tau}$ is always $\emptyset$, because the server never receives any block from the clients, which means that $\mathcal{U}^z_{j,i} = 1$ for any block b_j served by S. ■

Since S has to serve each block of the file at least once, we obtain the following corollary.

Corollary 1. *For at least β block transfers $\mathcal{U}^z_{j,i} = 1$.*

Lemma 4. *If there exists a host H that is receiving its first block in a time slot τ, then there is at least one block b_j in τ such that $\mathcal{U}^z_{j,i} = 1$.*

Proof: The number of active hosts in slot τ is $|\mathcal{I}^z_\tau|$. At most $|\mathcal{I}^z_\tau| - 1$ blocks can be transferred in τ because host H cannot upload to anyone. Then, since $d = u$, there exists at least one host H' that is on only for uploading. Let b_j be the block served by H'. As it is not downloading any block, $\mathcal{S}^z_{H',\tau} = \emptyset$ and hence $\mathcal{U}^z_{j,i} = 1$. ■

Corollary 2. *There are n hosts that receive a block for the first time. Thus, for at least n block transfers $\mathcal{U}^z_{j,i} = 1$.*

We now prove the claim. In order to compute the minimum energy consumption, we need to lower bound Equation 3.13. From Lemma 2, it follows that

$$\sum_{i=0}^{n-1}\sum_{j=0}^{\beta-1} \Delta_i \cdot \mathcal{D}^z_{j,i} = \beta \cdot \sum_{i=0}^{n-1} \Delta_i. \tag{4.4}$$

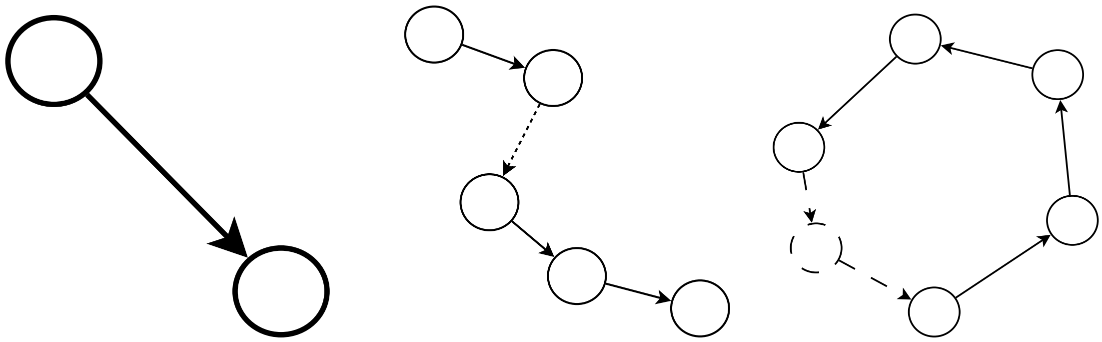
From Lemma 3 and Corollaries 1 and 2,

$$\sum_{i=0}^{n-1}\sum_{j=0}^{\beta-1} \Delta_{serv(j,i)} \cdot \mathcal{U}^z_{j,i} \geq \beta \cdot \Delta_S + \max\{0, n-\beta\} \cdot \min\{\Delta_S, \Delta_0\}. \tag{4.5}$$

Adding Equations 4.4 and 4.5, the claim follows. ■

The main implication of this result is that an optimal scheme should minimize the number of blocks transferred by the server. In order to complete the distribution of file, at least β blocks have to uploaded by the server. Optimal algorithms ensure that exactly β blocks are uploaded by the server. It is also worth noticing that, in this case, it does not matter whether the hosts have different power consumption. The transfer graphs in an optimal schedule look like as shown in Fig. 4.1.

It may also be noted that the order in which the hosts appear in the above transfer graphs does not matter. Also, the kinds of transfer graphs shown in Fig. 4.2 cannot be part of optimal schemes. We will see it in later chapters that these kinds of block transfer graphs may be used for more complicated cases where $k > 1$.

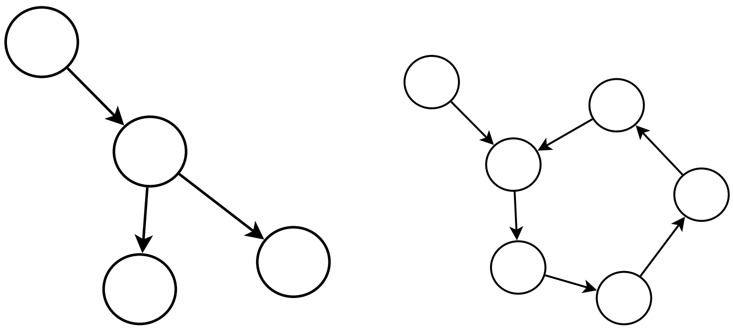


(a) A special kind of tree slot in which the server S serves a block to a host.

(b) A more general tree slot in which a chain to of block transfers is there.

(c) A cycle. Optimal kind of slot, this is what each optimal scheme maximizes.

Figure 4.1: Three kinds of slots that are seen in algorithms presented in this chapter. Note that each host is uploading (downloading) to (from) maximum one host.



(a) Optimality puts restriction that no host can upload to more than one.

(b) Optimality limits download from multiple hosts.

Figure 4.2: These kinds of graphs cannot be part of optimal schemes for $\frac{d}{u} = 1$.

4.2. Optimal Distribution Schemes

We now present optimal schemes achieving the lower bound of Theorem 4. We distinguish among three cases, depending on the relation between n and β, and we indicate the resulting schemes as Algorithms 1, 2 and 3. Note that in the pseudocode of algorithms, the transfer of block b_j from host H to host H' is expressed as $H \xrightarrow{j} H'$. While the three algorithms could be merged into one, we have chosen to present them separately for clarity.

Theorem 5. *When $d = u$, Algorithms 1, 2 and 3 describe optimal distribution schemes, where*

- *the energy consumed is* $E(z) = \beta\left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + \max\{0, n - \beta\} \min\{\Delta_S, \Delta_0\}$,
- *each host is on exactly β slots, except H_0 that is on $\max\{\beta, n\}$ slots, and*

- *no host is switched on (and off) more than thrice (twice in Algorithms 1 and 2), including the initial switch on and final switch off.*

- *energy consumed by host i in Algorithms 1 and 2 is equal to $\beta \cdot \Delta_i$. In Algorithm 3, $H_{\min}$ consumes $n\Delta_{min}$ and everyone else consumes $\beta\Delta_i$.*

Intuition for the optimality of the algorithms: Refer to the Appendix for the detailed proof. We start with Algorithm 1 (see Fig. 4.3), which is the simplest of the three, since it assumes that the number of clients is equal to the number of blocks. In the first n slots of the algorithm, the server uploads a distinct block of the file to each of the n clients. Since $n = \beta$, the server can upload the whole file to the clients in n slots. Then the server goes off. At this point, each host has a different block and needs to get the remaining $n - 1$ blocks. Then, in each of the remaining $n - 1$ slots, each client chooses another client to serve in a way that the resulting transfer graph is a cycle of the n hosts. In particular, each host i uploads the latest block it has received to host $i - 1$. This process continues for the next $n - 1$ slots, until all the hosts have all the blocks.

Algorithm 2 (Fig. 4.4), which assumes $n < \beta$, is more involved, but uses similar ideas as Algorithm 1. In Algorithm 3 (Fig. 4.5), the number of clients is larger than the number of blocks. Thus some hosts will have to upload the same block more than once. In this algorithm, once the server has served β distinct blocks, the host with the smallest energy consumption per slot uploads block b_0 to those other client without receiving any block.

$$S \xrightarrow{b_0} H_0 \qquad \begin{array}{c|ccc} S & b_0 & b_1 & b_2 \\ \hline H_0 & S & 0 & 0 \\ H_1 & 0 & 0 & 0 \\ H_2 & 0 & 0 & 0 \end{array}$$

(a) Slot 0: The server S, who has all the three blocks initially, uploads block 0 (b_0) to host 0 (H_0).

$$S \xrightarrow{b_1} H_1 \qquad \begin{array}{c|ccc} S & b_0 & b_1 & b_2 \\ \hline H_0 & S & 0 & 0 \\ H_1 & 0 & S & 0 \\ H_2 & 0 & 0 & 0 \end{array}$$

(b) Slot 1: The server S uploads b_1 to H_1.

$$S \xrightarrow{b_2} H_2 \qquad \begin{array}{c|ccc} S & b_0 & b_1 & b_2 \\ \hline H_0 & S & 0 & 0 \\ H_1 & 0 & S & 0 \\ H_2 & 0 & 0 & S \end{array}$$

(c) Slot 2: The server S uploads b_2 to H_2. The server goes to sleep.

$$\begin{tikzcd} & H_2 \arrow[dl, "b_2"'] & \\ H_1 \arrow[rr, "b_1"'] & & H_0 \arrow[ul, "b_0"'] \end{tikzcd} \qquad \begin{array}{c|ccc} S & b_0 & b_1 & b_2 \\ \hline H_0 & S & H_1 & 0 \\ H_1 & 0 & S & H_2 \\ H_2 & H_0 & 0 & S \end{array}$$

(d) All the three hosts exchange blocks, forming a cycle.

$$\begin{tikzcd} & H_2 \arrow[dl, "b_0"'] & \\ H_1 \arrow[rr, "b_2"'] & & H_0 \arrow[ul, "b_1"'] \end{tikzcd} \qquad \begin{array}{c|ccc} S & b_0 & b_1 & b_2 \\ \hline H_0 & S & H_1 & H_1 \\ H_1 & H_2 & S & H_2 \\ H_2 & H_0 & H_0 & S \end{array}$$

(e) The three hosts form another cycle.

Figure 4.3: Example of Algorithm 1, for $n = 3$ and $\beta = 3$. The label on each arrow is the index of the block being served. After each slot, as shown in the transfer graph, the changed state is shown as well. In this example, each state is a 3×3 matrix. Each row represents the blocks received by a particular host and each element in a row is the host uploading the block to this particular host.

Algorithm 1 Optimal scheme for $\beta = n$

for slot $j = 0 : n - 1$
    $S \xrightarrow{j} H_j$
for slot $j = n : 2n - 2$
    for $i = 0 : n - 1$
        $H_i \xrightarrow{(i+j) \bmod n} H_{(i-1) \bmod n}$

$$S \xrightarrow{b_0} H_0$$

S	b_0	b_1	b_2	b_3
H_0	S	0	0	0
H_1	0	0	0	0
H_2	0	0	0	0

(a) Slot 0: The server S, who has all the four blocks initially, serves b_0 to H_0.

$$S \xrightarrow{b_1} H_1$$

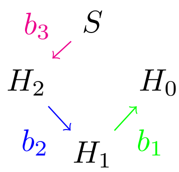
S	b_0	b_1	b_2	b_3
H_0	S	0	0	0
H_1	0	S	0	0
H_2	0	0	0	0

(b) Slot 1: The server S uploads b_1 to H_1.

$$S \xrightarrow{b_2} H_2$$

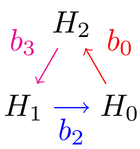
S	b_0	b_1	b_2	b_3
H_0	S	0	0	0
H_1	0	S	0	0
H_2	0	0	S	0

(c) Slot 2: The server S uploads b_2 to H_2.



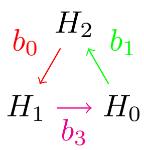
S	b_0	b_1	b_2	b_3
H_0	S	H_1	0	0
H_1	0	S	H_2	0
H_2	0	0	S	S

(d) Slot 3: All the hosts pass their block to next hosts, except the last one. Such slots occur $\beta - n$ times.



S	b_0	b_1	b_2	b_3
H_0	S	H_1	H_1	0
H_1	0	S	H_2	H_2
H_2	H_0	0	S	S

(e) Slot 4: Finally, when the server goes to sleep, they all cycle.



S	b_0	b_1	b_2	b_3
H_0	S	H_1	H_1	H_1
H_1	H_2	S	H_2	H_2
H_2	H_0	H_0	S	S

(f) Slot 5: The cycle continues until everyone has all the blocks.

Figure 4.4: Example of Algorithm 2, for $n = 3$ and $\beta = 4$.

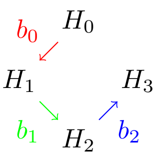
Algorithm 2 Optimal scheme for $\beta > n$

for slot $j = 0 : n - 1$
    $S \xrightarrow{j} H_j$
for slot $j = n : \beta - 1$
    $S \xrightarrow{j} H_{n-1}$
    for $i = 1 : n - 1$
        $H_i \xrightarrow{i+j-n} H_{i-1}$
for slot $j = \beta : \beta + n - 2$
    for $i = 1 : n$
        $H_{i \bmod n} \xrightarrow{(i+j-n) \bmod \beta} H_{i-1}$

$S \xrightarrow{b_2} H_2$

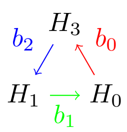
	S	b_0	b_1	b_2
H_0	S	0	0
H_1	0	S	0
H_2	0	0	S
H_3	0	0	0

(a) Slot 2: The server S, uploads b_2 to H_2. Initial slots are similar to Algorithm 1 and 2. The server goes to sleep.



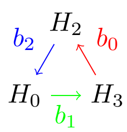
S	b_0	b_1	b_2
H_0	S	0	0
H_1	H_0	S	0
H_2	0	H_1	S
H_3	0	0	H_2

(b) Slot 3: β blocks are transfered from one to the next host. Such slots occur $n - \beta$ times.



S	b_0	b_1	b_2
H_0	S	H_1	0
H_1	H_0	S	H_3
H_2	0	H_1	S
H_3	H_0	0	H_2

(c) Slot 4: Cycling of hosts starts.



S	b_0	b_1	b_2
H_0	S	H_1	H_2
H_1	H_0	S	H_3
H_2	H_3	H_1	S
H_3	H_0	H_0	H_2

(d) Slot 5: Cycling continues. However, note that in each cycle, different set of hosts is involved.

Figure 4.5: Example of Algorithm 3, for $n = 3$ and $\beta = 4$.

Algorithm 3 Optimal scheme for $\beta < n$. $H_{\min}$ is the host with smallest Δ_i.

for slot $j = 0 : \beta - 1$
    $S \xrightarrow{j} H_j$
for slot $j = \beta : n - 1$
    $H_{\min} \xrightarrow{0} H_{j+1-\beta}$
    for $i = 1 : \beta - 1$
        $H_{i+j-\beta} \xrightarrow{i} H_{i+j+1-\beta}$
for slot $j = n : n + \beta - 2$
    $H_{2n-(j+1)} \xrightarrow{\beta-1} H_{n+\beta-(j+2)}$
    for $i = 0 : \beta - 2$
        $H_{(n+i-j) \bmod n} \xrightarrow{i} H_{(n+i-j-1) \bmod n}$

4.3. Fairness of Algorithms

During execution of Algorithm 1 and 2, each host is powered on exactly for β slots. Since we assume that switching on/off is free, energy consumed by host i is given as

$\beta \cdot \Delta_i, i \in \{S, 0, 1, 2, ..., n-1\}$, the best that can be done in terms of individual energy consumption because each host needs to be on for β slots to receive the whole file. All the hosts are on for exactly β slots to receive the file and the server is on for β slots to serve the file. Hence, these algorithms are fair to all the hosts as they all have equal contribution to energy consumption in the distribution process.

Algorithm 3 however, does not fall into the category of above algorithms, host H_0 is switched on for $(n-\beta)$ extra slots to upload to others. Other than it, all the other hosts are switched on exactly for β slots, implying that Algorithm 3 is unfair to the host with minimum power consumption as its energy consumption might be arbitrary larger (for a fixed β) than the others. Thus, energy E_0 consumed by host 0 is given by

$$E_0 = n \cdot \Delta_0$$

For the remaining hosts, energy consumption is given by

$$E_i = \beta \cdot \Delta_i, i \in \{S, 1, 2, ..., n-1\}$$

4.4. Optimal Number of Blocks in Homogeneous Systems

In the system model presented before, we had assumed that there are two sources of power consumption. In this section, we add a third source of power consumption along with the previous two, that is stated below:

- Each host consumes a fixed amount of energy $\delta_i \geq 0$, $i \in \mathcal{I}$ for each block served and/or received. This component δ_i captures the additional energy consumed by serving and receiving in the form of CPU activity [99], cooling, caching and hard disk activity, network card activity, etc. While in practice δ_i also depends on the size of the block, for simplicity, we assume that $\delta_i = \delta$, $\forall i \in \{S, 0, 1, \cdots, n-1\}$.

- For this section, we consider an energy-homogeneous system, in which all the hosts have the same energy consumption parameters, i.e., $P_i = P$ and $\delta_i = \delta$, for all $i \in \mathcal{I}$.

In this section, our goal is to find the optimal value of β, i.e., the number of blocks into which the file should be divided for minimum energy consumption. The number of blocks into which the file must be divided depends on the value of δ. If δ is very large, then it is better to divide the file in a small number of blocks, since each block transmission consumes additional energy δ because this will result in more number of transmissions, increasing the total energy consumed in transmissions.

On the other hand, if δ is small, we can divide the file into a number of blocks such that the energy consumed is reduced due to concurrent transfers. The following theorem summarizes the result.

Theorem 6. *In an energy-homogeneous system (i.e., $P_i = P$, $\forall i$) with $k = \frac{d}{u} = 1$, the value of β that minimizes the energy consumption of an optimal scheme is*

$$\beta = \min\left\{\sqrt{\frac{PB}{u\delta}}, n\right\} \tag{4.6}$$

Proof: From Theorems 4 and 5, the energy consumption of an optimal scheme z in an energy homogeneous system is

$$E(z) = (n\beta + \max\{n, \beta\}) \cdot \left(\frac{PB}{u\beta} + \delta\right) \tag{4.7}$$

To find the optimal value of β, we need to minimize the right hand side of Equation 4.7. Using Equation 4.7, we can rewrite Equation 3.13 as a function of β and δ as,

$$E(\beta) = \begin{cases} \dfrac{PB}{u}(n+1) + \delta(n+1)\beta, & \beta \geq n \quad (4.8) \\ \dfrac{nPB}{u}\left(1 + \dfrac{1}{\beta}\right) + \delta n(\beta + 1), & \beta \leq n \quad (4.9) \end{cases}$$

Note that in Equation 4.8 the first term is a constant and the second is linear in β. This is a straight line with positive slope $\delta(n+1)$. Hence, the function attains the minimum at the lower extreme $\beta = n$, where it intersects Equation 4.9. Hence it is enough to consider Equation 4.9 for $\beta \leq n$. Minimizing Equation 4.9 with respect to β we get,

$$\beta = \sqrt{\frac{PB}{u\delta}}. \tag{4.10}$$

When this value is larger than n the value $\beta = n$ has to be used. Note that if the value of $\sqrt{\frac{PB}{u\delta}}$ is not an integer, it has to be rounded to one of the two closest integer values, such that $E(\beta)$ is minimum. ■

Chapter 5

Optimal Solutions: Upload $>$ Download Capacity

The previous chapter introduced the basic idea behind the optimality of algorithms for a very simple yet realistic case. In this chapter, we focus on a scenario in which the upload capacity of the hosts is an integral multiple their download capacity, i.e., $u = kd, k \geq 2$. This typically happens for data centers for whom the data that they upload is much more than what they download.

The main findings reported in this chapter are that increasing upload to download ratio increases the energy efficiency of the overall system as compared to the algorithms presented in the previous chapter. We prove lower bounds on the energy under these circumstances and we also design algorithms that achieve the lower bound.

Before proceeding to the lower bounds when $u = kd$, we find similarities with the lower bounds presented for the case $d = u$. It can be easily seen that no matter how high is the upload capacity, the total data transfered in a slot will depend only on the total download capacity of the system, i.e., the total number of hosts (who are still to download at least one block) that are on in a slot. Hence, if h hosts are on in a slot, which do not have the complete file, then at most h blocks can be transfered in that particular slot.

More precisely, from Lemma 1, we can conclude that each host should download from at most one host in a time slot. However, due to the relationship between u and d, each host can upload to a maximum of k hosts.

5.1. Lower Bound

The following theorem provides a lower bound on the energy consumed by any distribution scheme when $k > 1$. Since $u > d$, the length of time slot is given as

$$\tau_{duration} = \frac{s}{d} \tag{5.1}$$

This clearly implies that each client has to be active for at least β slots to receive the whole file, whereas the server has to be active for at least β slots to upload one copy of each block to the clients. The next theorem immediately follows.

Theorem 7. *The energy required by any scheme z to distribute a file divided into β blocks among n clients when $k = \frac{d}{u} > 1$, satisfies*

$$E(z) \geq \lceil \frac{\beta}{k} \rceil \cdot \Delta_S + \beta \cdot \sum_{i=0}^{n-1} \Delta_i \tag{5.2}$$

Proof: From Eq. 5.1 we know that each host has to be active for at least β slots to receive the complete file. Since $u > d$ for all the hosts, from Lemma 1, we know that each host must download from at most one host. The server, however, can upload to k different hosts. It needs to make at least $\lceil \frac{\beta}{k} \rceil$ tree slots. ■

The main implication of this result is that an optimal scheme should minimize the number of blocks transferred by the server. In order to complete the distribution of a file, at least β blocks have to uploaded by the server. Optimal algorithms ensure that exactly β blocks are uploaded by the server. Hence, in case of $u > d$, from energy perspective, the only advantage is that server can upload to multiple hosts but the rest of the hosts need not use this high ratio of upload to download capacity. It is so because utilizing $u = 2d$ is sufficient for all the clients to achieve optimal energy consumption.

5.2. Optimal Distribution Schemes

Before beginning the formal treatment to the algorithms, we show an example of the basic idea behind the algorithms in this chapter (Fig. 5.1). Initially, the server serves to as many hosts, as permitted by the upload to download capacity ratio. Once it has disseminated all the blocks to the hosts, they cycle.

We now proceed to the presentation of the algorithms and their properties. The following theorem summarizes it.

Theorem 8. *When $u = kd$, Algorithms 4, 5, 6, 7 describe optimal distribution schemes, where*

1. *energy consumed is $E(z) = \lceil \frac{\beta}{k} \rceil \cdot \Delta_S + \beta \cdot \sum_{i=0}^{n-1} \Delta_i$*

2. *each host is on exactly β slots, except the server that is on $\lceil \frac{\beta}{k} \rceil$ slots*

3. *no host is switched on (and off) more than twice including the initial switch on and final switch off.*

4. *energy consumed by host i is equal to $\beta \cdot \Delta_i$, except by the server which consumes $\lceil \frac{\beta}{k} \rceil \cdot \Delta_S$*

Proof: It is trivial to see that the Algorithms achieve the lower bound as the server is active exactly for $\lceil \frac{\beta}{k} \rceil$ slots which are the only tree slots. The rest of the slots are slots with a cycle. Each host is switched off only after it receives the first block, after which no host is switched on/off until it receives the whole file. ∎

Theorem 9. *If each host can upload to at most one host. Then the case $u \geq kd$ is same as $u = d$. Also if a host can upload to multiple hosts and $u = d$ then uploading to one host is optimal and Algorithm 1 is optimal.*

Proof: From Lemma 1, the upload speed used by all the hosts is $u = d$ since no host can upload to more than one host. If a host can upload to at most one host at a time, then only two kinds of slots are possible, a tree slot and a slot with exactly one cycle. Hence, all the transmissions are equivalent to case $u = d$ because the uploading host cannot upload at a rate greater than $u = d$. ∎

$$\begin{tikzcd} S \arrow[r, "b_0"'] \arrow[d, "b_1"'] & H_0 \\ H_1 & \end{tikzcd}$$

S	b_0	b_1	b_2
H_0	S	0	0
H_1	0	S	0
H_2	0	0	0

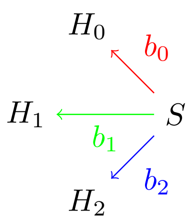
(a) The server can serve at most two blocks in a slot.

$$S \xrightarrow{b_2} H_2$$

S	b_0	b_1	b_2
H_0	S	0	0
H_1	0	S	0
H_2	0	0	S

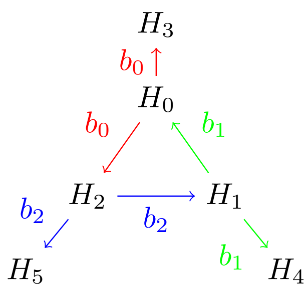
(b) The server serves only one block.

Figure 5.1: Example of Algorithm 7 when $u = 2d$, for $n = 3$ and $\beta = 3$.



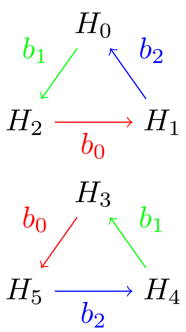
S	b_0	b_1	b_2
H_0	S	0	0
H_1	0	S	0
H_2	0	0	S
H_3	0	0	0
H_4	0	0	0
H_5	0	0	0

(a) The server can serve at most three blocks in a slot.



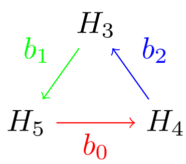
S	b_0	b_1	b_2
H_0	S	H_1	0
H_1	0	S	H_2
H_2	H_0	0	S
H_3	H_0	0	0
H_4	0	H_1	0
H_5	0	0	H_2

(b) Some hosts upload to multiple hosts.



S	b_0	b_1	b_2
H_0	S	H_1	H_1
H_1	H_2	S	H_2
H_2	H_0	H_0	S
H_3	H_0	H_4	0
H_4	0	H_1	H_5
H_5	H_3	0	H_2

(c) Two different slots because two different transfer graphs.



S	b_0	b_1	b_2
H_0	S	H_1	H_1
H_1	H_2	S	H_2
H_2	H_0	H_0	S
H_3	H_0	H_4	H_4
H_4	H_5	H_1	H_5
H_5	H_3	H_3	H_2

(d) The three hosts form another cycle.

Figure 5.2: Example of how Algorithm 5 works when $u = 3d$, for $n = 6$ and $\beta = 3$.

We provide the intuition of how the algorithms are working through two examples that we elaborate next. Algorithm 7 presents an optimal scheme for the trivial case when $u = kd$ and $n \leq 2$. It is worthwhile noticing that having $u > 2d$ is of no use in this case as the maximum number of nodes to which any host can upload is two. Note that in the first slot, the server has three slots to serve and it can serve at most two blocks because $u = 2d$. In the next slot, only one block is to be served, it serves. The rest of the slots are as described in the previous chapter. All the hosts form a cycle to serve the blocks to each other Fig. 5.1. The same idea works for any $n \geq \beta$.

An example of Algorithm 5 is provided in Fig. 5.2. In the first slot, the server serves all the three blocks and switches off. In the next slot, the fact that $u > d$ is used and three hosts upload to two hosts. All the hosts form a cycle to serve the blocks to each other.

Algorithm 4 Optimal scheme for trivial cases of $u = kd$, $n > \beta$

Arrange $P_0 \leq P_1 \cdots \leq P_i \cdots \leq P_{n-1}$
if $\beta = 1$ **then**
  $S \rightarrow P_0, \cdots, P_{k-1}$ in one slot
  P_0 serves k hosts until last-1 slot
  P_0 serves $(\beta \bmod k)$ in last slot
end if
if $\beta = 2$ **then**
  for slot $j = 0$
  if $(n \mod 2 = 1)$ & $k > 2$ **then**
    $S \xrightarrow{0} H_0, S \xrightarrow{1} \{H_1, H_{n-1}\}$
  else
    $S \xrightarrow{0} H_0, S \xrightarrow{1} H_1$
  end if
  for slot $j = 1 : \lfloor \frac{n-1}{2} \rfloor$
    $H_{2j-2} \xrightarrow{0} H_{2j-1}, H_{2j}$
    $H_{2j-1} \xrightarrow{1} H_{2j-2}, H_{2j+1}$
  slot++;
  if $n \mod 2 = 1$ **then**
    $H_{2j-2} \xrightarrow{0} H_{2j-1}, H_{2j}$
    $H_{2j-1} \xrightarrow{1} H_{2j-2}$
    if $k = 2$ **then**
      slot++;
      $H_0 \xrightarrow{1} H_{n-1}$
    end if
  else
    $H_{2j-2} \xrightarrow{0} H_{2j-1}, H_{2j}$
    $H_{2j-1} \xrightarrow{1} H_{2j-2}, H_{2j+1}$
    slot++;
    $H_{2j-2} \xrightarrow{0} H_{2j-1}$
    $H_{2j-1} \xrightarrow{1} H_{2j-2}$
  end if
end if

Algorithm 5 Optimal scheme for case $u = kd$, $n > (\beta \geq 3)$

for slot $j = 0 : \lfloor \frac{\beta}{k} \rfloor - 1$
  for $i = 0 : k - 1$
    $S \xrightarrow{j \cdot k+i} H_{j \cdot k+i}$
for slot $j = \lfloor \frac{\beta}{k} \rfloor$
  for $i = 0 : \beta \bmod k - 1$
    $S \xrightarrow{j \cdot k+i} H_{j \cdot k+i}$
for round $\rho = 0 : \lfloor \frac{n}{\beta} \rfloor - 2$
  for slot $j = 0 : \beta - 2$
    for $i = 0 : \beta - 1$
      $H_{\rho \cdot \beta+i} \xrightarrow{(i+j) \bmod \beta} H_{\rho \cdot \beta+(i-1) \bmod \beta}$
      if $j = 0$
      $H_{\rho \cdot \beta+i} \xrightarrow{(i+j) \bmod \beta} H_{(\rho+1) \cdot \beta+i}$
if $n \mod \beta = 1$ **then**
  if $\beta = 3$ **then**
    slot 1:$H_2 \xrightarrow{2} H_1$, $H_1 \xrightarrow{1} \{H_0, H_3\}, H_0 \xrightarrow{0} H_2$
    slot 2:$H_2 \xrightarrow{0} H_1$, $H_3 \xrightarrow{1} H_2, H_1 \xrightarrow{2} H_3$
    slot 2:$H_0 \xrightarrow{0} H_3$, $H_3 \xrightarrow{2} H_0$
  else
    for slot $j = 0 : \beta - 3$
      for $i = 0 : \beta - 1$
        $H_{\rho \cdot \beta+i} \xrightarrow{(i+j) \bmod \beta} H_{\rho \cdot \beta+(i-1) \bmod \beta}$
        $H_{(\rho+1)\beta-1-j} \xrightarrow{\beta-1-j} H_{(\rho+1) \cdot \beta}$
    for slot $j = \beta - 1$
      for $i = 2 : \beta$
        $H_{\rho \cdot \beta+i} \xrightarrow{i-2} H_{\rho \cdot \beta+i-1}$
      $H_1 \xrightarrow{1} H_{(\rho+1) \cdot \beta}$
    for slot $j = \beta$
        $H_{\rho \cdot \beta} \xrightarrow{0} H_{(\rho+1) \cdot \beta}$
        $H_{(\rho+1) \cdot \beta} \xrightarrow{\beta-1} H_{\rho \cdot \beta}$
  end if
else
  for round $\rho = \lfloor \frac{n}{\beta} \rfloor - 1$
    for slot $j = 0 : \beta - 2$
      for $i = 0 : n \mod \beta - 1$
        $H_{\rho \cdot \beta+i} \xrightarrow{(i+j) \bmod \beta} H_{\rho \cdot \beta+(i-1) \bmod \beta}$
        $H_{\rho \cdot \beta+i} \xrightarrow{(i+j) \bmod \beta} H_{(\rho+1) \cdot \beta+i}$
      for $i = n \mod \beta : \beta - 1$
        $H_{\rho \cdot \beta+i} \xrightarrow{(i+j) \bmod \beta} H_{\rho \cdot \beta+(i-1) \bmod \beta}$
  ρ++
  begin slot
      for $i = 0 : n \mod \beta - 1$
        $H_{\rho \cdot \beta+i} \xrightarrow{i} H_{\rho \cdot \beta+i+1}$
      $H_{n-1} \xrightarrow{\beta-1} H_{\rho \cdot \beta}$
  end slot
end if

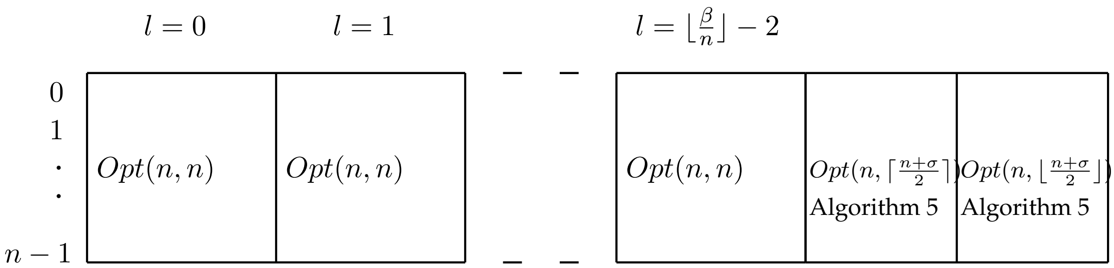


Figure 5.3: A representation of optimal algorithm for $\beta > n$ when $u = kd$,i.e., Algorithm 7 using basic cycling and Algorithm 5. Observe that this is optimal.

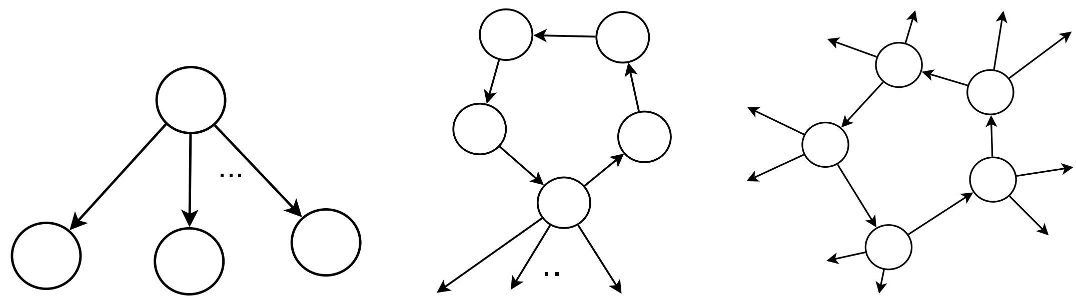
Algorithm 6 Optimal scheme for trivial cases for $u = kd$, $\beta \geq n$

Arrange $P_0 \leq P_1 \cdots \leq P_i \cdots \leq P_{n-1}$
if $n = 1$ **then**
  Server serves all the blocks in $\beta - 1$ slots
end if
if $n = 2$ **then**
  for slot $j = 0 : \lfloor \frac{\beta}{2} \rfloor - 1$
    $S \xrightarrow{2j} H_0, \;\; S \xrightarrow{2j+1} H_1$
  if $\beta \bmod 2 = 1$ **then**
    for slot $j = \lfloor \frac{\beta}{2} \rfloor$
      $S \xrightarrow{2j} H_0, \;\; S \xrightarrow{2j} H_1$
  end if
  for slot $j = 0 : \lfloor \frac{\beta}{2} \rfloor - 1$
    $H_0 \xrightarrow{2j} H_1, \;\; H_1 \xrightarrow{2j+1} H_0$
end if

Algorithm 7 can be visualized in the form of basic cycling, i.e., Algorithm 1 and Algorithm 5, as shown in Fig. 5.3. In the basic cycling, server serves all the diagonal blocks to the hosts. In the last, the number of blocks are divided such that they can be solved using Algorithm 5. Note that in the last two blocks shown in the figure, the number of blocks are such that they are always greater than or equal to the number of hosts. Since, it is not optimal for hosts to receive from multiple hosts in this case, each block in this figure can be completed one by one without consuming extra energy.

Algorithm 7 Optimal scheme for case $u = kd$, $\beta \geq (n \geq 3)$

Let $\beta = \alpha \cdot n + \sigma, n \geq 4, \beta > 1, \sigma < n, k \leq \lceil \frac{n+1}{2} \rceil$
for slot $j = 0 : \lfloor \frac{(\alpha-1)\cdot n}{k} \rfloor - 1$
  for $i = 0 : k - 1$
    $S \xrightarrow{j\cdot k+i} H_{(j\cdot k+i) \bmod n}$
for slot $j = \lfloor \frac{(\alpha-1)\cdot n}{k} \rfloor + 1 : \lfloor \frac{\beta}{k} \rfloor - 1$
  for $i = 0 : k - 1$
    $S \xrightarrow{j\cdot k+i} H_{(j\cdot k+i) \bmod \lceil \frac{n+\sigma}{2} \rceil}$
for slot $j = \lfloor \frac{\beta}{k} \rfloor$
  for $i = 0 : (\beta \bmod k) - 1$
    $S \xrightarrow{j\cdot k+i} H_{(j\cdot k+i) \bmod \lfloor \frac{n+\sigma}{2} \rfloor}$
for round $r = 0 : \alpha - 2$
  for slot $j = 0 : n - 2$
    for $i = 0 : n - 1$
      $H_i \xrightarrow{r\cdot n+i} H_{(i-1) \bmod n}$
if $\sigma = 0$ **then**
  for round $r = \alpha - 1$
    for slot $j = 0 : n - 2$
      for $i = 0 : n - 1$
        $H_i \xrightarrow{r\cdot n+i} H_{(i-1) \bmod n}$
else
  if $n = 3$ **then**
    if $\sigma = 1$ **then**
      $H_0 \xrightarrow{3} H_2 \xrightarrow{2} H_1 \xrightarrow{1} H_0$
      $H_0 \xrightarrow{0} H_2 \xrightarrow{3} H_1 \xrightarrow{2} H_0$
      $H_1 \xrightarrow{1} H_2 \xrightarrow{0} H_1$
    else
      $H_2 \xrightarrow{2} H_1 \xrightarrow{1} H_0 \xrightarrow{3} H_2$
      $H_2 \xrightarrow{3} H_1 \xrightarrow{4} H_2$
      $H_2 \xrightarrow{2} H_0 \xrightarrow{0} H_1 \xrightarrow{1} H_2$
      $H_2 \xrightarrow{4} H_0 \xrightarrow{0} H_2$
    end if
    Let $\beta_1 = \lceil \frac{n+\sigma}{2} \rceil, \beta_2 = \lfloor \frac{n+\sigma}{2} \rfloor$
    Call Algorithm 5 for (n, β_1)
    Call Algorithm 5 for (n, β_2)
  end if
end if



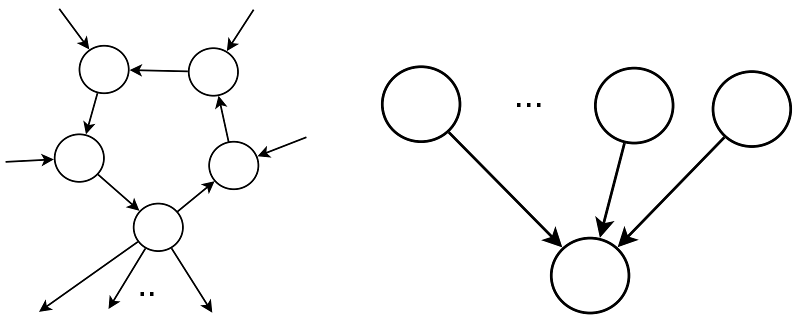
(a) One host uploading to at most k hosts at their full download capacity.

(b) One host in a cycle, can still upload to at most k.

(c) All the hosts in a cycle uploading to $m < k$ hosts.

Figure 5.4: We assume $u = kd$, for sufficiently high k. Users uploading to multiple hosts in various situations.

Fig. 5.4 presents examples of the block transfer graphs that can be part of an optimal scheme in $u = kd$ case. A host can upload to multiple hosts depending on k but note that it can receive from only one (Lemma 1). On the other hand, Fig. 5.5 presents the kinds of block transfer graphs that cannot be part of an optimal scheme in the case $u = kd$.



(a) (b)

Figure 5.5: We assume $u = kd$. Users are receiving from multiple hosts.

These block transfer graphs cannot be part of any optimal scheme because they have hosts which are receiving from multiple hosts.

In this case, the system is limited by download capacity. Hence, uploading simultaneously to many hosts does not help. It does help to reduce the number of tree slots made by the server initially. Thus, scheme with $k = n$ minimizes the number of tree slots and the number of tree slots increase as k decreases, reducing the energy efficiency of the schemes.

Chapter 6

Optimal Solutions: Download $>$ Upload Capacity

The last two chapters have introduced optimal algorithms for $u = kd, k \in \{1, 2, 3, \cdots\}$. In this chapter, we focus on a scenario in which the download capacity of the hosts is an integral multiple their upload capacity, i.e., $d = ku, k \geq 2$. This is the most common case in the residential connections in the Internet today.

The findings in this chapter are more interesting as compared to the previous chapters because the fact that a host can receive from multiple hosts in a slot gives us different kinds of transfer graphs that were not possible until now. It is also worth emphasizing that since a host can receive from multiple hosts, hosts with high power consumption can receive the whole file in less than β time slots, which was not possible in the earlier cases. Thus, increasing download to upload ratio increases the energy efficiency of the overall system. We also note that if all the hosts have the same power consumption, then $\frac{d}{u} = 2$ is good enough to yield optimal results. We compute lower bounds on the energy under these circumstances and also design algorithms that achieve the lower bound.

Before proceeding to the lower bounds when $d = ku$, we find similarities with the lower bounds presented for the case $d = u$. It can be easily seen that no matter how high is the download capacity, the total data transfered in a slot will depend only on the total upload capacity of the system, i.e., the total number of hosts that are on in a slot. Hence, if h hosts are on in a slot, then at most h blocks can be transfered in that particular slot.

Lemma 5. *Given that $u_{\max} \leq d_{\min}$, restricting the upload to only one host at a time does not increase the total energy consumption.*

Proof: Consider a set of the hosts $\mathcal{L} = \{0, 1, 2, .., l\},\ \ l \geq 2, P_i > 0,\ \ \forall i \in \mathcal{L}$. Let us assume that host i with upload capacity u_i is serving to the rest of the hosts. We show that serving individually at full capacity to different hosts consumes energy no more than serving hosts in parallel, for any i.

Let the size of block be s, the time required to upload to all the hosts in parallel is at least $\frac{ls}{u_i}$. Thus, the energy consumed in complete transfer is

$$E_{parallel} = P_i \frac{ls}{u_i} + \left(\sum_{j=1, j \neq i}^{l} P_j \right) \frac{ls}{u_i} \tag{6.1}$$

In case H_i uploads the block to the hosts sequentially one by one, energy consumed is

$$E_{sequential} = P_i \frac{ls}{u_i} + \left(\sum_{j=1, j \neq i}^{l} P_j \right) \frac{s}{u_i} < E_{parallel}, \ \ \forall i \in \mathcal{L} \tag{6.2}$$

■

We define three kinds of scenarios depending on the energy consumption of the hosts:

- *Energy homogeneous scenario:* If all the hosts including the server, have equal power consumption, i.e., $P_S = P_i = P \ \forall i \in \{0, 1, \cdots, n-1\}$ holds, then we call it as homogeneous scenario.

- *Restricted energy homogeneous scenarios:* All the hosts may not have equal power consumption but the highest power consumption of any host is bounded by the average of the power consumption of the other hosts. We emphasize that this is the most practical scenario.

- *Energy heterogeneous scenario:* Each host i has power consumption P_i.

6.1. Energy Homogeneous Scenario

In this setting, the possibility to download more than one block in a slot implies that the minimum number of slots in which a host has to be on can possibly be less than β. It is so because a host can receive multiple blocks in one slot. This possibility makes it difficult to come up with a lower bound in all the cases and hence design optimal algorithms as well.

6.1.1. Lower Bound

For the homogeneous case, Equation 3.13 reduces to

$$E(z) = \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} c_{j,i}^{z} = P \cdot \frac{s}{u} \cdot \sum_{i=0}^{n-1} \sum_{j=0}^{\beta-1} \left(\mathcal{D}_{j,i}^{z} + \mathcal{U}_{j,i}^{z} \right) \tag{6.3}$$

Thus, to minimize the energy consumption for any scheme z, minimize,

$$\sum_{i=0}^{n-1}\sum_{j=0}^{\beta-1}(\mathcal{D}_{j,i}+\mathcal{U}_{j,i})$$

The following two theorems characterize the behavior of optimal schemes in a homogeneous setting for any $k \geq 2$.

Theorem 10. *A scheme for homogeneous system is optimal if and only if it minimizes the number of tree slots.*

Proof: A scheme is optimal $\Rightarrow$ It minimizes the number of tree slots.

Consider a scheme $z \in \mathcal{Z}_k^{n,\beta}$ that has T tree slots. Also assume that z finishes in S slots. Let the cost of slot τ be c_τ^z if n_τ blocks are transfered in it. Note that there can be only two kinds of slots in normal schemes, either a slot with a cycle or a tree slot. If slot τ is a tree slot, then

$$c_\tau^z = (n_\tau + 1)\cdot\Delta \tag{6.4}$$

If slot τ is slot with a cycle, then

$$c_\tau^z = n_\tau\cdot\Delta \tag{6.5}$$

So the cost of z, $c(z)$ is given by

$$c(z) = \sum_{\tau=1}^{S} c_\tau^z = \left(\sum_{\tau=1}^{S} n_\tau + T\right)\cdot\Delta = (n\beta + T)\cdot\Delta \tag{6.6}$$

Since z is optimal, $n\beta$ is the total number of blocks to be transfered. Clearly, $c(z)$ is minimized for $T = T_{min}$ for any k and given n and β.

If a scheme minimizes the number of tree slots $\Rightarrow$ The scheme is optimal.

It is trivial to see that if a scheme z minimizes the number of tree slots then its energy consumption can be given by $c(z) = (n\beta + T_{min})\cdot\Delta$. No scheme that has a lesser cost can exist because $n\beta$ is the number of blocks that must be transfered and T_{min} is the minimum number of tree slots possible. So z must be optimal. ■

Theorem 11. *Let z be an optimal schedule in an energy homogeneous system such that $d = ku$. Then the energy consumed by z satisfies*

$$n\beta\cdot\Delta \leq E(z) \leq (n\beta + \max\{n,\beta\})\cdot\Delta \tag{6.7}$$

Proof: The upper bound follows directly from Theorem 10. The maximum number of tree slots can be $\max\{n,\beta\}$. The lower bound follows if we assume that there are no tree slots, the best possible case.

■

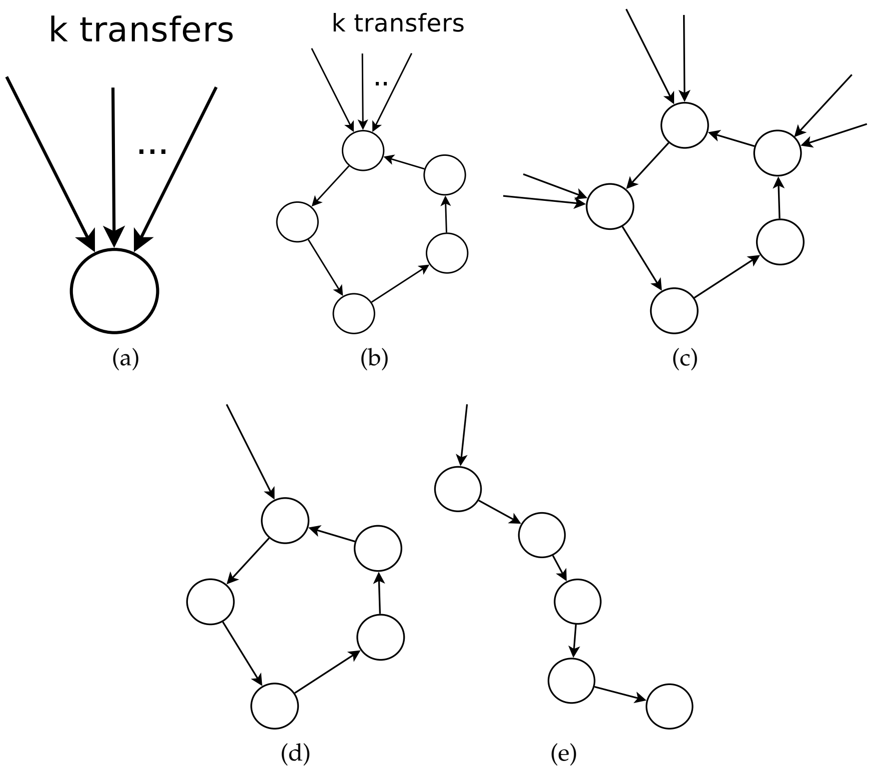


Figure 6.1: Different kinds of slots possible in this case as $d = ku$.

Fig. 6.1 shows some of the allowed block transfer graphs for a homogeneous system with $d = ku$. While in principle all the slots are possible, we conjecture that in an optimal scheme only the last two kinds of slots are possible. It is so because such slots increase the parallelism in the system and more blocks can be downloaded in parallel. The first three kinds of slots are more suited in the energy heterogeneous scenario so that the hosts with extremely high power consumption can be turned off by uploading blocks to them at the maximum rate.

6.1.2. File Distribution Schemes

Note that the lower bound on energy consumption when $\beta \leq n$ presented in Theorem 11 is the same as the lower bound presented in Theorem 4 for $k = 1$, when applied to energy homogeneous systems. The energy consumption of Algorithms 1 and 3 in an energy homogeneous system with $\beta \leq n$ is exactly $n(\beta + 1)\Delta$ (Theorem 5). Hence, Algorithms 1 and 3, which were optimal for $\beta \leq n$ in case of $k = 1$ are optimal in this case as well.

However, if $\beta > n$, the algorithm for $k = 1$ (Algorithm 2) is not optimal anymore if $k > 1$. So we present Algorithm 8 and Algorithm 9, that describes distribution schemes for this case. Note that both these schemes use $k = 2$ only.

6.1.3. Quasi-Optimal Distribution Schemes

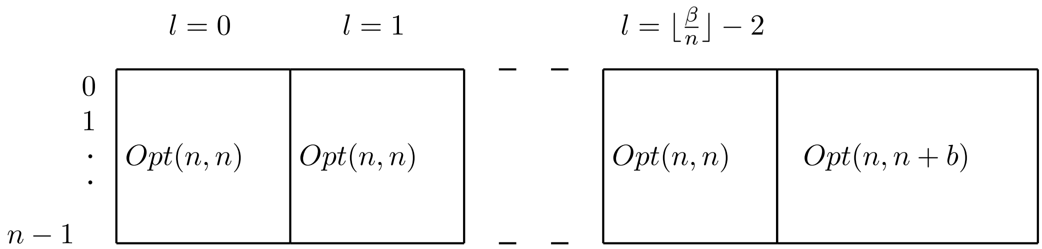


Figure 6.2: A representation of Algorithm 8 to visualize the distribution of blocks using Algorithm 1 and 2.

Algorithm 8 distributes the file among the clients using ideas from Algorithms 1 and 2. We represent the state of process with a two dimensional array A of size $n \times \beta$ (Fig. 6.2) with the rows and the columns representing the clients and the blocks, respectively. We set an entry $A_{ij} = 1, i \in \{0, 1, .., n-1\}, j \in \{0, 1, .., \beta - 1\}$ if and only if H_i has received b_j, and 0 otherwise. At the beginning, all the entries are 0 and after the completion of the algorithm they all should be 1. Furthermore, imagine the array A divided in $\lfloor \frac{\beta}{n} \rfloor - 1$ square subarrays of size $n \times n$ and one rectangular subarray of size $n \times (n + b)$. $Opt(n, \beta)$ indicates the optimal algorithm corresponding to the values of n and β. Note that this is just a conceptual division to understand Algorithm 8 in terms of Algorithms 1 and 2.

After the first loop, the diagonal of the first square subarray is set to 1, i.e., $A_{ii} = 1, \forall i \in \{0, ..., n-1\}$. Additionally, after the second loop, the top left corner position (see Fig. 6.2) of each subarray has also been set to 1, i.e., $A_{0j} = 1, \forall j \in \{0, n, 2n, .., (\lfloor \frac{\beta}{n} \rfloor - 1)n\}$. In each iteration of the for loop at Line 10, the elements of one of the subarrays of $n \times n$ are set to 1 by serving in the same fashion as in Algorithm 1, while the server completes serving the diagonal of the next square/rectangular subarray. When Line 45 is reached, all the elements of all the square subarrays are marked as 1. The remaining blocks are served using Lines 3-9 of Algorithm 2, with an appropriate relabeling of the blocks.

Theorem 12. *In a homogeneous system with $k > 1$,*

- *If $\beta \leq n$, then Algorithm 1 (when $\beta = n$) and Algorithm 3 (when $\beta < n$) describe optimal distribution schemes with energy $E(z) = n(\beta + 1) \cdot \Delta$ (Thm 11)*

- *If $\beta > n$, let $b = \beta \bmod n$, then Algorithm 8 describes a distribution scheme with energy*

$$E(z) = \left(n(\beta + 1) + \left\lfloor \frac{\beta}{n} \right\rfloor + b - 1 \right) \cdot \Delta \tag{6.8}$$

Algorithm 8 Energy saving scheme for case $k = 2$ and $\beta > n$

$b = \beta \bmod n$
for $j = 0 : n - 1$
  begin slot
  $S \xrightarrow{j} H_j$
  end slot
for $j = 1 : \lfloor \frac{\beta}{n} \rfloor - 1$
  begin slot
  $S \xrightarrow{nj} H_0$
  end slot
for $l = 0 : \lfloor \frac{\beta}{n} \rfloor - 2$
  for $j = 0 : n - 2$
    begin slot
    $S \xrightarrow{(l+1)n+j+1} H_{j+1}$
    for $i = 0 : n - 1$
      $H_i \xrightarrow{ln+((i+j) \bmod n)} H_{(i-1) \bmod n}$
    end slot

- *If $\beta > n$, then Algorithm 9 describes a distribution scheme with energy*

$$E(z) = \left(n(\beta + 1) + \left\lceil \frac{2\beta}{n(n-1)} \right\rceil \right) \cdot \Delta \tag{6.9}$$

- *In a homogeneous system with $k > 1$, Algorithm 8: Energy consumed by S is $\beta\Delta_S$, host 0 is $(\left\lfloor \frac{\beta}{n} \right\rfloor n + b)\Delta$ and by H_i, $\forall i \in \{1, 2, .., n-1\}$ is $(\lfloor \frac{\beta}{n} \rfloor (n-1) + b + 1) \cdot \Delta$. Thus, this algorithm is unfair to host 0 which consumes $(\lfloor \frac{\beta}{n} \rfloor - 1) \cdot \Delta$ more energy compared to the other hosts. Additionally, no host is switched on (and off) more than thrice.*

Proof: Refer Appendix. ∎

Note that for $\beta \leq n$, Algorithms 1 and 3 are still optimal even though $k > 1$. This indicates that increasing k does not always result in energy savings. The fact that $k > 1$ is helpful only when the number of blocks is greater than the number of hosts. The intuition is that if a host receives from, say, $k \geq 2$ hosts, it happens at the cost of at least $k - 1$ hosts who cannot receive, because the upload degree of a transfer graph is limited by the number of nodes. This essentially nullifies the effect of parallel uploads to a host in this scenario where all the hosts have equal power consumption.

While Algorithm 8 does not achieve optimal energy when $\beta > n$, it is quasi-optimal (in addition to asymptotically optimal), since it is off from the lower bound by an additive term of $(\lfloor \beta/n \rfloor + b - 1)\Delta$, which is usually much smaller than the term $n(\beta + 1)\Delta$.

Algorithm 9 improves Algorithm 8. It is off from the lower bound by an additive term of $O(\frac{\beta}{n^2})$. We conjecture that this is the best any algorithm can achieve in the case $d = ku$ for homogeneous system. The additional additive term signifies that there will be one tree slot every $O(n^2)$ slots as compared to one tree slot per $O(n)$ slots in Algorithm 8. Both the algorithms are however, close to optimal and implementation of Algorithm 8 is much simpler than that of Algorithm 9.

Algorithm 9 Optimal scheme for case $d = ku$

for slot $j = 0 : n - 1$
  $S \xrightarrow{j} H_j$
while $(\beta > 1 + \sum_{i=1}^{\text{var}} (n - i))$ **do**
    var++
end while
var = var -1
for $(\xi = 1; \xi \leq \text{var}; \xi + +)$ **do**
  for slot $j = \sum_{i=1}^{\xi} (n - i) + 1 : \min \left\{ \sum_{i=1}^{\xi+1} (n - i), \beta - 1 \right\}$
    $S \xrightarrow{j} H_{n-1}$
    $H_0 \xrightarrow{j - \sum_{i=1}^{\xi} (n-i) - 1} H_{n-\xi}$
    for $i = 1 : n - 1$
      $H_i \xrightarrow{(i+j-n) \bmod \beta} H_{i-1}$
end for
$\xi = 1$
dif=0
for slot $j = \beta : \boldsymbol{var} + \sum_{i=1}^{\boldsymbol{var}+1} (n - i)$
if $\xi = \text{var} \;\&\; \left(\sum_{i=1}^{\text{var}+1} (n - i) - (\beta - 1) \right) \neq 0$ **then**
    for $i = 1 : n - \xi$
      $H_i \xrightarrow{(i+j-n) \bmod \beta} H_{i-1}$
    $H_0 \xrightarrow{\min\left\{ \sum_{i=1}^{\xi+1} (n-i), \beta-1 \right\} - \sum_{i=1}^{\xi} (n-i) + \text{dif}} H_{n-\xi}$
  dif++
else
    for $i = 1 : n - \xi$
      $H_i \xrightarrow{(i+j-n) \bmod \beta} H_{i-1}$
    $H_0 \xrightarrow{\min\left\{ \sum_{i=1}^{\xi+1} (n-i), \beta-1 \right\} - \sum_{i=1}^{\xi} (n-i)} H_{n-\xi}$
  ξ++
end if
for slot $j : n + \beta - 2$
  $H_0 \xrightarrow{(j - \text{var}) \bmod \beta} H_{n-\text{var}-1}$
  for $i = 0 : n - \text{var}$
    $H_i \xrightarrow{(i+j-n) \bmod \beta} H_{i-1}$

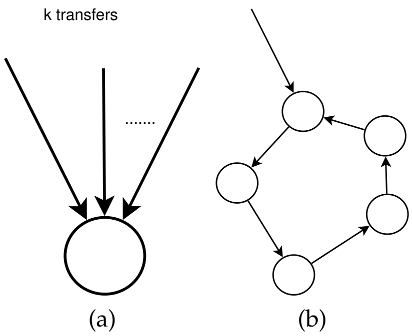


Figure 6.3: Equation 6.10 distinguishes between these two kinds of slots.

6.2. Restricted heterogeneous scenario

In practice, all the hosts may not have the same power consumption. This section discusses the most practical scenario where the power consumption of participating hosts may lie. Throughout this section, we assume that the hosts are numbered in increasing order of their power consumption, i.e., $P_0 \leq P_1 \leq \cdots \leq P_{n-1}$. The algorithms presented earlier in this chapter, namely, Algorithm 8 and 9 are energy efficient provided,

$$P_j \leq \frac{\sum_{i=0}^{j-1} P_i}{j-1} \tag{6.10}$$

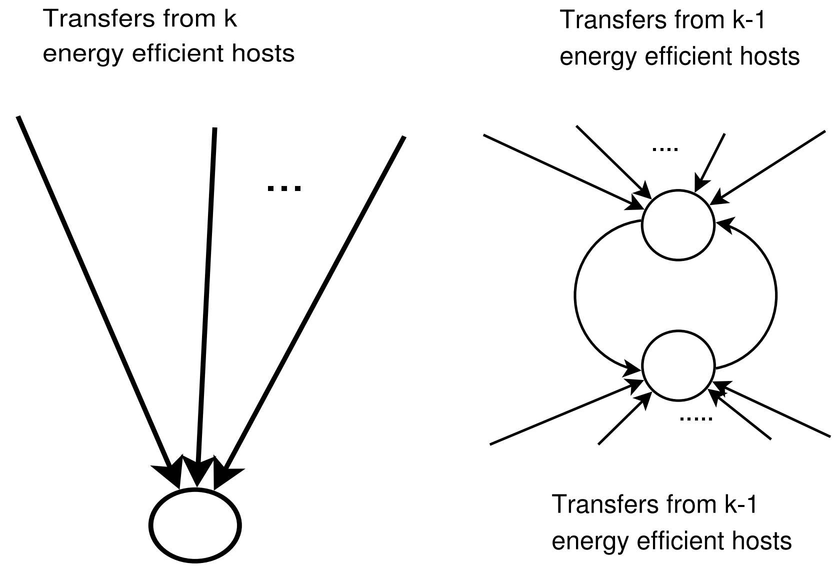
The interpretation of Equation 6.10 is that the host with maximum power consumption is such that it is more energy efficient for hosts to make slots with cycle rather than tree slots in which a host receives multiple blocks from multiple hosts in one slot. Fig. 6.3 clarifies this argument. The equation basically puts a condition on P_i's so that a slot with cycle is less power hungry than a tree slot in which one is being served by many hosts. This allows us to continue using the algorithms that were designed for homogeneous scenario.

6.3. Heterogeneous scenario

In this section, we keep no restriction on power consumed by host i. For the first few hosts, if they follow Equation 6.10, then Algorithms 8 and 9 can be used for them. However, for the hosts whose power consumption does not fit in any of these equations, multiple downloads from low power hosts can be done.

6.3.1. Energy efficient slots

The sum of the costs of all the blocks transferred during slot τ of duration $\frac{s}{u}$ is equal to the cost of that slot, i.e., $\sum_{i\in\mathcal{I}_\tau^z}\sum_{b_j\in\mathcal{S}_{i,\tau}^z} c_{j,i}^z = c_\tau^z$. Hence, the energy consumed by the



(a) Instead of just receiving one block in a slot, power hungry host receives k blocks from energy efficient k hosts.

(b) Two power hungry hosts cycle. In the same slot, they are being served by energy efficient hosts as well.

Figure 6.4: The above two kinds of transfer graphs are added to the list of slots in optimal schemes.

scheme is

$$E(z) = \sum_{i=0}^{n-1}\sum_{j=0}^{\beta-1} c^z_{j,i} = \frac{s}{u} \cdot \sum_{i=0}^{n-1}\sum_{j=0}^{\beta-1} \left(P_i \cdot \mathcal{D}^z_{j,i} + P_{serv(j,i)} \cdot \mathcal{U}^z_{j,i}\right) \tag{6.11}$$

Theorem 13. *Any scheme can be converted to a scheme that has only tree slots and a slot with a cycle and also with lesser energy consumption.*

Proof: For any scheme, the first slot has to be tree slot. Now we need to prove that no slot in optimal scheme can have more than one cycle. For more than one cycle, clearly at least one of the hosts need to upload to multiple hosts. For any such slot, there exists a slot that makes the same transfer and consumes lesser energy (Lemma 5). ■

Corollary 3. *Optimal scheme can consist of only tree slots and slots with exactly one cycle. No other kinds of transfer graphs can be part of any optimal scheme.*

Fig. 6.4 adds a new kind of transfer slot to the previously known set.

Algorithm 10 presents an algorithm for case $d = ku$ for energy heterogeneous scenario. In this, the first few hosts can finish downloading the whole file in a batch if they are energy efficient. Once all the energy efficient hosts have downloaded, they upload the power hungry hosts at their full capacity. This done for only two hosts and those two hosts make use of their upload capacities as well by forming a cycle among each other.

Algorithm 10 Optimal scheme for case $d = ku$

1: Arrange $P_i's$ as $P_0 \leq P_1 \cdots \leq P_{i-1} \leq P_i \cdots \leq P_{n-1}$
2: First serve optimally as allowed by the restricted heterogeneous strategies, if the number of such hosts is more than $2k - 2$
3: Serve optimally to first $2k - 2$ hosts using strategies mentioned in restricted heterogeneous scenario
4: From there on take two hosts, the first tree slots are of the kind shown in Fig. 6.4a
5: The remaining slots with cycle can be done as Fig. 6.4b
6: Repeat until all the hosts have the file

Chapter 7

Performance Analysis

The theory presented in the earlier chapters provide a useful insight on the problem. We do a numerical analysis of the theoretical results to enhance our understanding. In order to assess the performance of our schemes, we have run an extensive simulation study with two objectives:

1. To quantitatively evaluate the results of our analysis.

2. To compare our results with the existing results on energy efficient file distribution techniques, P2P in particular.

The findings of this chapter are detailed in the rest of the Chapter. Section 7.1 describes the network and energy model. It also introduces the other approaches to which we compare our schemes and the metric used to compare all the schemes. Section 7.2 considers the schemes for which we assume that all the hosts have different power consumption. In the end, this section also provides the road map for the experiments presented in the chapter. Section 7.3 considers the energy homogeneous scenario in which we compare the energy consumed by the algorithms designed in the Thesis among themselves. We also compare the optimal algorithms in the Thesis to various other energy efficient approaches. We also detail the impact of block sizes and upload capacities on the energy consumed. In the end, we compare the performance of the algorithms when energy consumed in switching on/off is not zero. Impact of load dependent hardware is also studied.

7.1. Experimental Setup

In our experiments, we assume that each host can send messages to all the other hosts, i.e., we consider the topology of a complete graph. This is the topology of many application-level overlays made of Internet hosts. We do not consider the intermediate devices like switches or routers in our evaluation, since we assume that they cannot be turned off at will.

7.1.1. Network Model

At the host level, we assume continue with the assumptions made in theory that all the hosts have equal upload and download capacity during the file distribution. Specifically, we have considered the following values for the relevant input parameters in our experiments: upload and download capacity $u = d = 10$ Mbps. Finally, unless otherwise stated, we consider a scenario with one server and 1000 hosts. This homogeneous network scenario models a corporate network in which both the network infrastructure and the whole set of devices belong to the same company/organization, and are centrally managed.

7.1.2. Energy Model

We consider two kinds of energy consumption scenarios: *Energy Homogeneous Scenario* and *Energy Heterogeneous Scenario*. In energy homogeneous scenario, power consumed by all the hosts is taken to be 80 W. On the other hand, in energy heterogeneous scenario, all the hosts may have different power consumption.

For our experiments in *Energy Homogeneous Scenario*, we consider two different energy models. In the first one, the hosts only have two power states: an *OFF* state, in which they do not consume power, and an *ON* state, in which they consume the full nominal power, equal to 80 W (typical nominal power consumption for notebooks and desktop PCs lies in the range 60 to 80 W [97] [100]). Unless otherwise stated, this is the default energy model for our experiments.

We also consider an enhanced energy model in order to understand the impact of load proportional energy consumption in our schemes. We consider a model that fits most of the current network devices [97], in which the energy consumed has some dependency on the CPU utilization and network activity [101]. This energy model is characterized by four states. Besides the OFF state, the other states are: the *IDLE* state, in which the device is active but not performing any task, and consuming 80% of the nominal power; and the *TX-or-RX* state, in which the device is active and either transmitting or receiving, and consuming 90% of the nominal power; the *TX-and-RX* state, in which the device is active and both transmitting and receiving, and consuming its full nominal power. We considered this model to analyze the impact of load proportionality on the overall energy consumption of the schemes considered in our experiments.

In the heterogeneous scenario, we analyze the effect of having devices with heterogeneous power consumption profiles. For this purpose we use the previously described two-state model, but we assume that for each host its nominal power consumption is randomly uniformly drawn from [60W, 120W].

Although power consumed is different for multicore processors and architecture [102], we do not make this distinction and we consider the values for single core processors. In our model, similar results are expected if we change the architecture to multicore.

7.1.3. Schemes and Metric for Comparison

We consider *six* file distribution schemes in the performance evaluation. They are described below:

- *Opt:* The file distribution schemes presented throughout the Thesis.

- *Parallel:* A scheme in which all the users download the same file at the same time from the same server in parallel. This is one of the most common architectures for file distribution.

- *Serial:* A scheme in which the server uploads the complete file in sequence to the hosts involved in the file distribution process. That is, the server uploads the complete file to the first host who switches on to receive the file, once it finishes, it switches off. Then the server uploads the file to the second host, which switches on to receive the file and switches off after receiving it and so on until all the hosts have the file.

- *BitTorrent:* The most commonly used protocol for P2P file sharing. For details of the algorithm used, refer to [103].

- *P2Proxy:* A scheme presented in [42] as a method to reduce energy consumed by P2P file sharing. In this scheme, a dedicated proxy server downloads the files on behalf of the users inside a corporation. Once downloaded the file, proxy sends the whole file to each host one by one. Note that it is different for *Serial* . If there are c corporations with each having h number of hosts, then each corporation will have a proxy which will download using the legacy bittorent. Once it downloads it sends it to the rest of the h hosts in the corporation using *Serial* .

- *Balancing:* A class of schemes presented in [31] for energy efficient P2P file sharing. In this scheme, hosts having high $\frac{u_i}{P_i}$ ratio are uploaded first by the server and then these hosts aid the server in transferring the file to the rest of the hosts.

The **metric** we have used in order to compare the energy consumption of different file distribution schemes is *energy per bit*, computed as the ratio of the total amount of energy consumed by the distribution process, divided by the total number of bits transfered using a particular scheme.

7.1.4. Road Map for the Experiments

In the rest of the chapter, we assume that all the hosts have the same upload and download capacity. We divide our analysis based on heterogeneous and homogeneous energy scenario. All the shown results have

In heterogeneous energy scenario, we first plot the three basic schemes *Opt* , *Serial* and *Parallel* . By doing this, we realize that the energy consumed per bit changes with the number of hosts for *Serial* and *Parallel* . In contrast, it remains constant for *Opt* and depends only on the file size. This helps us fix the number of hosts to 1000 in the rest of the experiments. After this we, start discussing homogeneous energy scenario. To start with, we compare the algorithms designed in the Thesis for various cases depending on the relationship between upload and download capacity. We choose one of the algorithms designed in the Thesis to compare with the most relevant P2P approaches. After fixing the algorithm to compare with, we also see the impact of load dependent hardware and on/off energy on our algorithms.

All the graphs are shown such that the X-axis represents file size in bits and the Y-axis represents energy per bit with unit *Joule per bit*. The X-axis has a total of 296 points for all the experiments. Hence, to minimize the overlapping of points, we show at most two points per curve so that the graphs can be visualized in a better manner.

7.2. Heterogeneous Energy Scenario

In order to validate the analysis, in Fig. 7.1 we have plotted the energy per bit consumed by the file distribution process as a function of the size of the file, for the three different file distribution schemes, namely, *Opt*, *Serial* and *Parallel*. As we can see, our schemes perform consistently better than both serial and parallel schemes. In particular, by maximizing the amount of time in which hosts serve while being served, our schemes tend towards reducing orders of the magnitude of the total energy cost with respect to *Parallel*. The energy savings with respect to *Serial* scheme can be as high as 50%. This performance improvement with respect to the *Serial* scheme is due to the use of P2P-like distribution, and indeed it decreases as the file size (and the number of blocks into which it is split) decreases.

Moreover, we can also observe how the *Parallel* scheme performs consistently worse than any other scheme, consuming orders of magnitude more than the *Serial* and *Opt* schemes. Since the utilization of both these schemes is widespread in the current Internet, our observations confirm the great potential of distributed schemes for saving energy. Fig. 7.1 also depicts the performance of our *Opt* algorithm for different number of hosts (50, 500, and 5000). We observe that the energy per bit consumed by *Opt* as well as *Serial* is not affected by the number of hosts in the scheme. Note that the energy con-

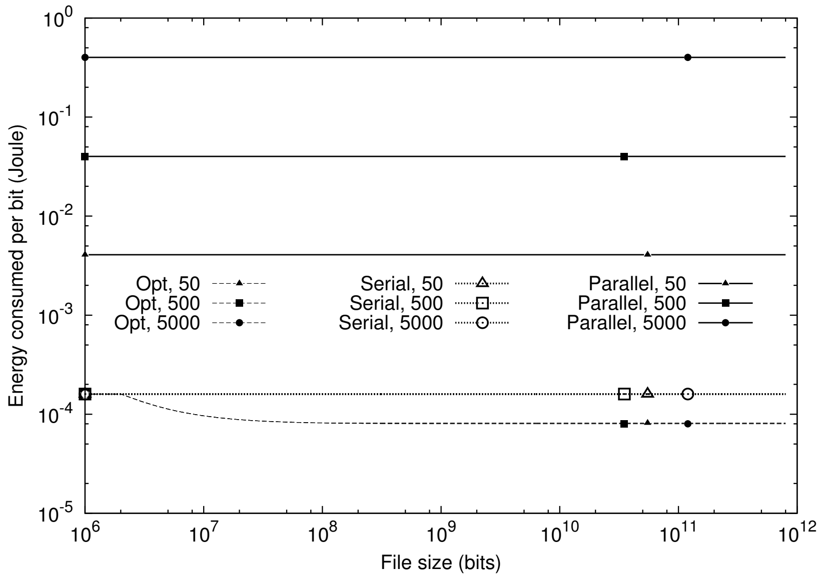


Figure 7.1: Comparison of *Serial* and *Parallel* with *Opt* . The legends represent the name of the scheme and the number of hosts considered. As described before each curve has 296 points but only two are shown. Block size: 256kB.

sumed by *Opt* and *Serial* is independent of the number of hosts. The same is however, not true for *Parallel* . The initial overlap of *Opt* and *Serial* is due to the fact that file is too small to be divided into multiple blocks. In this case, both, *Opt* and *Serial* consume the same energy.

It is clear from this analysis that the number of hosts is immaterial for *Opt* . Hence, for the rest of the Chapter, unless otherwise stated, we will present results exclusively for a setting with 1000 hosts.

In the current state of the art, where many energy efficient solutions for P2P file distribution exist, it is not fair to compare with the schemes that are energy agnostic. To be able to evaluate our schemes more fairly, we compare them with *P2Proxy* and *Balancing* schemes as well.

7.3. Energy Homogeneous Scenario

In this section, we assume that all the hosts have the same power consumption. As we have seen in the previous section that if the variation between power consumption of the devices is not huge, the qualitative behavior of the algorithms is the same.

7.3.1. Comparison between Optimal schemes

Until now, in theory we had been solving three different cases for energy efficiency. We first evaluate three different schemes corresponding to the three different situations depending on the relation between upload and download capacity.

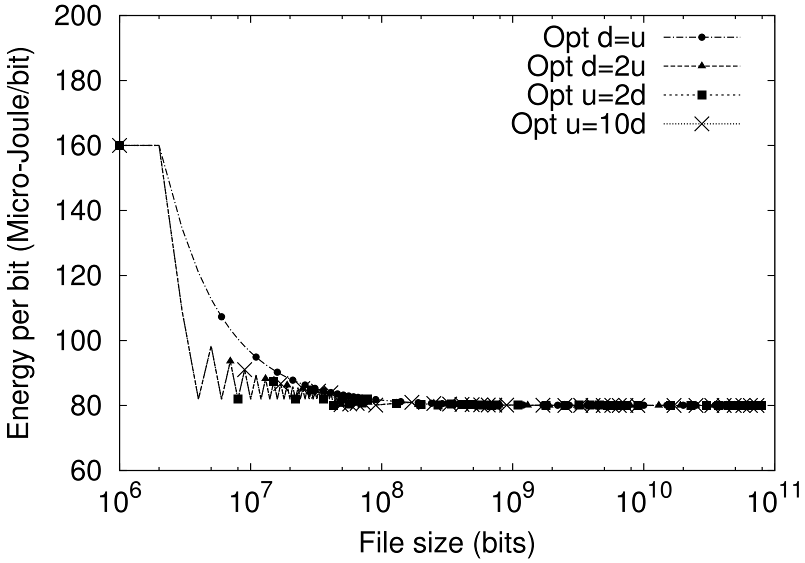


Figure 7.2: The optimal schemes in three scenarios, i.e., $d = u$, $u = kd$ and $d = ku$ in the energy homogeneous scenario. Number of hosts = 1000. Block size: 256kB.

Fig. 7.2 displays four different schemes that we have designed throughout the Chapters 4-6. It may be noted that there is no substantial difference between four schemes if we consider only large files. Having high upload to download ratio or download to upload ratio, is more energy efficient for smaller files. Usually software updates and music files fall into this category. Since these files are shared very routinely, the asymmetry in upload and download capacities is useful in achieving more energy gains. Thus, it can be concluded that for large files, it does not really help to have anything other than $d = u$. A simple explanation to this is that all the files are if a total of $n\beta$ blocks are to be downloaded, of which only few can be in more energy efficient manner by making use of high k. For example, in case of $u = kd$, only server is able to use $k > 1$, for the rest of the hosts, the best strategy is to stick to $d = u$. Similarly for $d = 2u$ majority of the hosts use only $d = u$.

All the schemes coincide initially because when the number of blocks is just one, all the algorithms are essentially the same.

7.3.2. Comparison with Other Energy Efficient Proposals

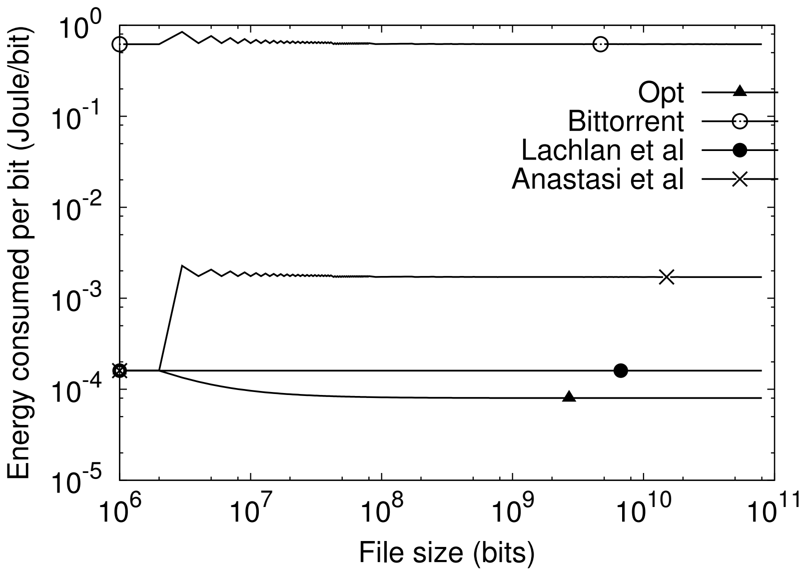


Figure 7.3: Comparison with other energy efficient P2P schemes and BitTorrent. Number of hosts: 1000. Block size: 256kB.

Fig. 7.3 presents a comparison between *Opt* with $d = u$ other proposals in a energy homogeneous system. We observe that the bittorent without energy efficiency is very energy expensive. The other two approaches considered, improve the energy consumption of bittorent and already achieve three to four orders of magnitude improvement in energy savings. Our algorithm is the best among all these that provides 50% improvement over the next best *Balancing* and more than an order of magnitude compared to *P2Proxy*. To compute the energy consumed by *P2Proxy*, we assume that there are 50 different corporations each having 20 hosts. Thus, fifty hosts participate in P2P file distribution. Once they receive the file they send to their hosts.

The three energy efficient schemes coincide when the file is divided in only one block. In this case, all the energy efficient schemes upload each block to each host one by one. Only the hosts uploading and downloading are kept on. However, this is not true for bittorent, since all the hosts are on the energy consumption is high even when the file consists of just one block. This increases the energy consumption per bit for bittorent.

7.3.3. Energy Consumption for Different Block Sizes

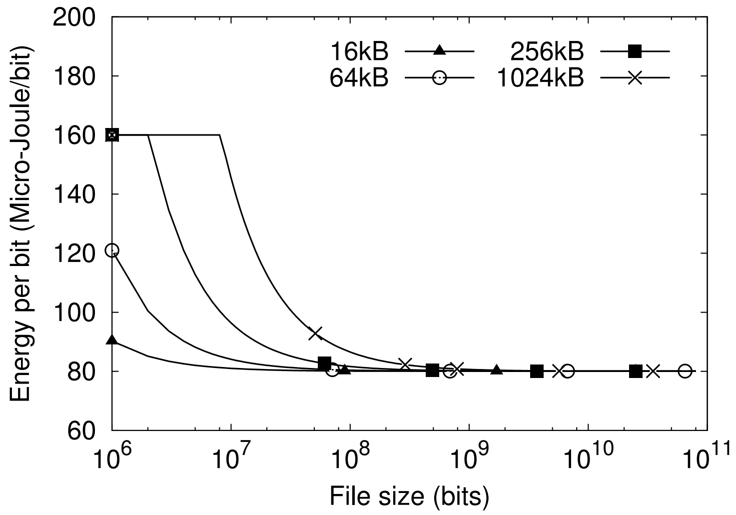


Figure 7.4: Energy consumption per bit for different block sizes. Number of hosts: 1000.

As the file size increases, impact of block size decreases for larger blocks because energy is higher for small files. It is so because lesser number of blocks will be there and distribution will be more sequential. However, as the file size increases, the parallelism in the optimal schemes is exploited and energy consumption is lowered.

7.3.4. Energy Consumption for Different Upload Capacities

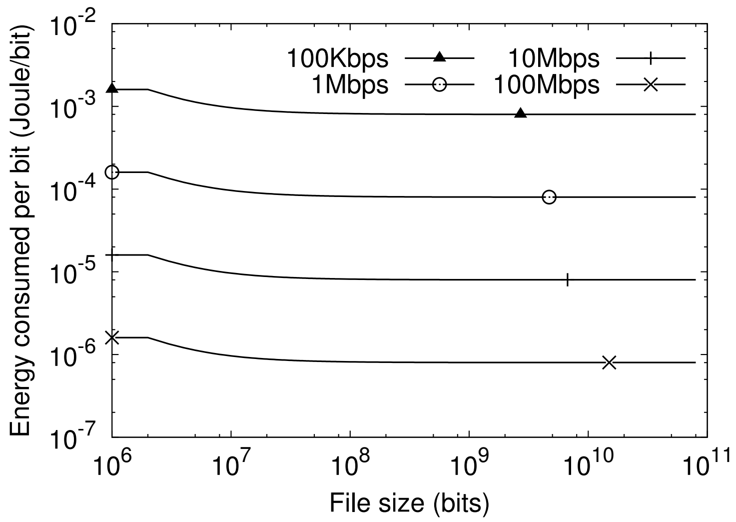


Figure 7.5: Energy consumption per bit for different upload speeds. Block size: 256kB. Number of hosts: 1000

It is clear that the capacities play big role in absolute energy consumption and this is intuitive because all the hosts have to be on for a longer period of time if the upload capacities are low. Hence, the higher the upload capacity, the lesser the energy consumed.

7.3.5. ON/OFF Energy Costs

As seen in Chapters 4-6, our optimal algorithms develop in rounds. Typically, not every host is on in every round (i.e., some go on and off more than once during the file distribution process). In a realistic scenario, a host takes some time to both go off (or into a very low power mode), and to get back to active mode. Usually, this on/off time is in the order of a few seconds [96]. The additional amount of energy consumed while switching between these power states (that we call here "on/off costs") has potentially an important impact on the energy performance of a scheme, penalizing specifically those schemes in which host activity is more "discontinuous" over time.

In order to mitigate the negative impact of on/off costs, in our simulations we implement the on/off costs and the result is as shown in Fig. 7.6.

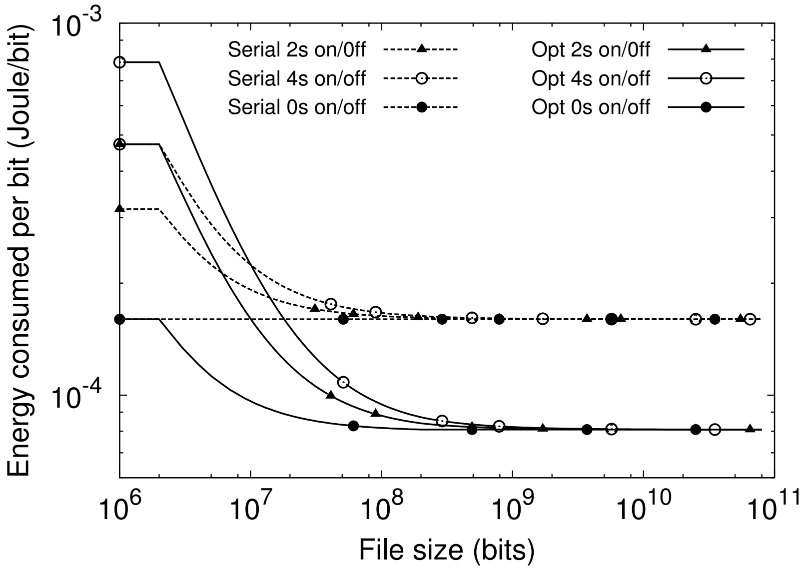


Figure 7.6: Impact of on/off energy cost on the energy per bit consumed by our algorithm, as a function of file size. Number of hosts: 1000. Block size: 256kB.

Fig. 7.6 presents the energy consumed by our scheme in comparison to the serial scheme considering a switch on/off time equal to 2 and 4s. As expected, the on/off costs increase the energy per bit consumed by all schemes. This increment is more pronounced for small file sizes, where we see that on/off costs make the performance of our scheme closer (but still better) to the serial scheme. Conversely, for medium/large file sizes, the contribution of on/off costs to the total energy consumed by a scheme becomes marginal, and the performance of both the optimal scheme and the serial approaches the one in the case without on/off costs.

7.3.6. Load Dependency

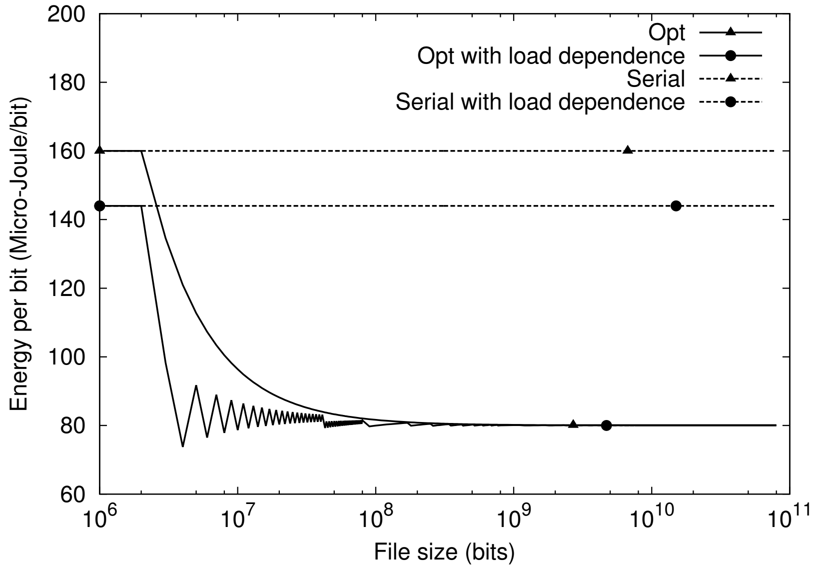


Figure 7.7: Impact of the energy model on the energy per bit consumed by our algorithm as a function of file size. Number of hosts: 1000. Block size: 256kB.

In this set of experiments, we have analyzed the impact of the four-state energy model described in Section 7.1.2, which implies some degree of energy proportionality of the host devices. The research community is investing a lot of effort in enabling energy proportionality. Hence, in the future it is expected that network devices will consume energy proportionally to the supported load. Fig. 7.7 shows the results assuming the four-state energy model. We see that the *Serial* scheme with load dependence performs consistently better than *Serial* without load dependence. It is so because in *Serial* , all the hosts are either involved in uploading or in downloading, never both. On the other hand, *Opt* optimizes the transfers in which both download and upload happen simultaneously. So if such transfer dominate the full power consumption dominates. From this, it is clear that if the costs of just upload/download becomes 50% of the cost with upload and download both, serial will do as good as optimal. However, it is unlikely to be so because power profiles are usually not impacted by so much by adding such a small overhead.

We also observe that there is a sharp initial decrease in the load balancing with optimal algorithm but it converges soon. The reason is that initially the blocks transfers are more serial than parallel and impact of energy savings from load-balancing is more likely for sequential transfers.

Chapter 8

Conclusions and Future Work

This Thesis presents contributions to the goal of reducing energy consumption in a file distribution process. It defines a model to capture energy consumption in file distribution that is particularly suited to the environments in which all the participating hosts are under the control of one administrative domain. We prove that the more general scenarios are NP-complete and coming up with optimal energy consumption is not feasible.

We design optimal algorithms for the distribution under the constraints in which upload or download capacities are integral multiples of each other, which is generally the case in the Internet. Hence, the algorithms presented in this Thesis are general enough to be applied to file transfers in various domains. For example, they can be used to distribute a file using peer to peer clients like Bittorrent. Software updates are an important part of ICT today [14] and can be an important application of our work. Initially the updates are with servers only and it has to reach multi-millions users. The files uploaded by the user to a cloud are replicated in many servers located in different places for fault tolerance reasons. The cache updates of content distribution networks, etc.

Apart from the application of the complete algorithms, even parts of our algorithms can be used in various scenarios. For example, if multiple servers have to synchronize, they all have 1 GB (for example) file to be distributed to each other. Our algorithms can be used in this scenario. The situation will be analogous to having received chunks from the server Fig. 1.5a. Now they all can distribute to each other as in the subsequent figures. A more realistic application can be that an amount of data is to be downloaded instead of a file. The users do it independently from a server. After some time they use our algorithms to share among themselves. This circumvents the strict scheduling required by our algorithms.

The message that we get from the Thesis is that from energy consumption perspective, having high upload/download or download/upload ratios do not reduce energy consumption for most of the scenarios. The only case in which having high download-

/upload ratio may help in reducing energy consumption is when the hosts have dramatically different power consumptions. Hence, in the most of the Thesis, we concentrate on schemes for which the ratios are at most 2.

Finally, through numerical analysis, we can deduce that our algorithms are optimal for large file sizes and produce results independent of the cost of switching between sleep and wake up states. Also, for large files the size of blocks in which the file has been divided does not play any significant role. If the hardware on which our algorithms run are energy proportional, our optimal algorithms converge.

Now we discuss the future work, some additional results, beyond what is presented in the Thesis but are desirable. Foremost, we would like to enhance the application of our algorithms to more general cases in which the upload and download capacities can assume any arbitrary values. Even though the problem is NP-complete in that case, it would be desirable to find some scenarios in which our algorithms still perform well.

In the Thesis, we assume that the hosts once start participating in the process, neither fail nor leave the process in the middle. Another dimension can be added to the Thesis by analyzing the impact of the hosts leaving in the middle of the file distribution process. Similar analysis may be done for the hosts that arrive late. But this is not so crucial as the late comers may be batched in the next round of the distribution of the same file.

Another important milestone is to actually prove that the algorithms designed for the case $d = ku$ are optimal as well. We already show that they are off by a very small margin but theoretically showing optimality will close the work nicely.

Finally, to make the problem more applicable, multiple files to be shared between multiple hosts have to be considered. It is still to be seen how the algorithms behave in this case.

Appendices

Appendix A

A.1. Proofs of Correctness and Optimality for $k = 1$

For the correctness and optimality proofs of a scheme z (described by an algorithm), we define the state $\sigma_{i,\tau}^{z}$ of a host $i \in \mathcal{I}$ at the end of slot τ as the set of blocks held by that time at the host. Thus, to start with, initially for S we have, $\sigma_{S,0}^{z} = \mathcal{B}$, and, for each client $i \in \{0, ..., n-1\}$, $\sigma_{i,0}^{z} = \emptyset$. If z is correct, after the makespan of z (τ_f^z slots) the state of every client $i \in \{0, ..., n-1\}$ must be $\sigma_{i,\tau_f^z}^{z} = \mathcal{B}$. We omit z and τ when clear from the context.

A.1.1. Algorithm 2

Let us denote the scheme described by Algorithm 2 as z_2. This scheme has the following properties.

Observation 1. *After the* for *loop at Lines 28-2,*
(i) the state of client i is $\sigma_i = \{b_i\}, \forall i \in \{0, .., n-1\}$, and
(ii) all hosts, including the server, have been switched on once and switched off once, except host H_{n-1}, which was only switched on once.

Lemma 6. *After the q^{th} iteration of the loop at Lines 3-6, for $q \in \{0, 1, .., \beta - n\}$, each host $H_i, i \in \{0, ..., n-1\}$, has state*

$$\sigma_i = \bigcup_{p=0}^{q} \{b_{(i+p)}\} \tag{A.1}$$

Proof: We use induction on q to prove the lemma. The base case ($q = 0$) follows from Observation 1.(i).

Induction step: Assume the hypothesis to be true for the $(q-1)^{\text{th}}$ iteration. Client $H_i, i \in \{0, ..., n-2\}$ receives block $b_{(i+q)}$ in the q^{th} iteration, while client H_{n-1} receives block $b_{(q+n-1)}$ from the server. Thus, $\forall i \in \{0, ..., n-1\}$, the state of client H_i after the q^{th} iteration is

$$\sigma_i = \bigcup_{p=0}^{q-1} \{b_{(i+p)}\} \cup \{b_{(i+q)}\} = \bigcup_{p=0}^{q} \{b_{(i+p)}\}$$

■

Lemma 7. *After the q'^{th} iteration of the loop at Lines 7-9, for $q' \in \{0, 1, .., n-1\}$, each host $H_i, i \in \{0, 1, .., n-1\}$, has state*

$$\sigma_i = \bigcup_{p=0}^{q'+\beta-n} \{b_{(i+p) \bmod \beta}\} \tag{A.2}$$

Proof: We use induction on q' to prove the claim. The base case ($q' = 0$) follows from Lemma 6 with $q = \beta - n$. Let the claim (induction hypothesis) be true for the $(q'-1)^{\text{th}}$ iteration. In the q'^{th} iteration, the value of j is $j = q' + \beta - 1$. Hence, H_i receives block $b_{(i+q'+\beta-n)}$. Thus, the state of client H_i after the q'^{th} iteration is

$$\sigma_i = \bigcup_{p=0}^{q'-1+\beta-n} \{b_{(i+p) \bmod \beta}\} \cup \{b_{(i+q'+\beta-n) \bmod \beta}\} = \bigcup_{p=0}^{q'+\beta-n} \{b_{(i+p) \bmod \beta}\} \tag{A.3}$$

■

Lemma 8. *During the execution of Algorithm 2 each host $H_i, i \in \{0, ..., n-1\}$ serves a block that it has already downloaded.*

Proof: Let us consider the loops at Lines 3-6 and Lines 7-9 in sequence. In the q^{th} iteration of these loops, host H_i serves block $b_{(i+q-1) \bmod \beta}$. From the previous lemmas, after the $(q-1)^{\text{th}}$ iteration of these loops, host H_i has state

$$\sigma_i = \bigcup_{p=0}^{q-1} b_{(i+p) \bmod \beta}$$

which includes $b_{(i+q-1) \bmod \beta}$. Hence the claim follows. ■

Theorem 14. *After the termination of Algorithm 2 each host H_i, $i \in \{0, ..., n-1\}$, has received all the blocks $b_j \in \mathcal{B}$ with optimal energy $E(z_2) = \beta(\Delta_S + \sum_{i=0}^{n-1} \Delta_i)$. Additionally, host i consumes exactly $\beta\Delta_i$ energy, and no host has been switched on (and off) more than twice.*

Proof: It follows from Lemma 7 that each host has received all the blocks at the end of the loop at Lines 7-9. Then, the scheme is correct since each host serves a block that it has already downloaded (Lemma 8). Each host (including the server) is active exactly β slots. Then, the total energy consumed is $E(z_2) = \beta(\Delta_S + \sum_{i=0}^{n-1} \Delta_i)$, which is optimal since it matches the lower bound. Each host is on for exactly β slots. Hence, the total energy consumed by host i is $\beta\Delta_i$.

It follows from Lemma 6 that all the hosts, including the server, have been on during the execution of the loop at Lines 3-6. Similarly, Lemma 7 means that all the hosts but the server have been on during the execution of the loop at Lines 7-9. This, together with

Observation 1.(ii), implies that the server and host H_{n-1} have been switched on (and off) once, whereas the rest of the hosts were switched on/off twice. ■

A.1.2. Algorithm 3

For the correctness and optimality proofs of Algorithm 3 we define the state $\zeta^z_{r,\tau}$ of a block b_r at the end of τ as the set of clients $H_i, i \in \{0, ..., n-1\}$, who have received b_r. Thus, to start with, $\forall r \in \{0, ..., \beta-1\}$, initially the state of block b_r is $\zeta^z_{r,0} = \emptyset$. After the makespan τ^z_f of scheme z, the state should be, $\forall r \in \{0, ..., \beta-1\}$, $\zeta^z_{r,\tau^z_f} = \bigcup_{i=0}^{n-1}\{H_i\}$

Let us denote the scheme described by Algorithm 3 as z_3. This scheme has the following properties.

Observation 2. *After the* for *loop at Lines 1-2,* $\forall r \in \{0, 1, .., \beta-1\}$*, the state of block* b_r *is* $\zeta_r = \{H_r\}$.

Lemma 9. *After the* q^{th} *iteration of the* for *loop at Lines 3-6, for* $q \in \{0, ..., n-\beta\}$*, the state of block* b_r *is*

$$\zeta_r = \bigcup_{p=0}^{q}\{H_{r+p}\} \tag{A.4}$$

Proof: We prove the claim using induction on q. The base case ($q = 0$) is trivially true by the observation. Assume the statement to be true for the $(q-1)^{\text{th}}$ iteration. In the q^{th} iteration, $q = j + 1 - \beta$. Then, block b_r is served to H_{r+q}. Thus, the state of block b_r after the q^{th} iteration is

$$\zeta_r = \bigcup_{p=0}^{q-1}\{H_{r+p}\} \cup \{H_{r+q}\} = \bigcup_{p=0}^{q}\{H_{r+p}\}$$

■

Lemma 10. *After the* q'^{th} *iteration of the* for *loop at Lines 7-10, for* $q' \in \{0, 1, .., \beta-1\}$*, the state of block* b_r *is*

$$\zeta_r = \bigcup_{p=0}^{n-\beta}\{H_{r+p}\} \bigcup_{p=0}^{q'}\{H_{(r-p) \bmod n}\} \tag{A.5}$$

Proof: The base case ($q' = 0$) is true from Lemma 9 after the loop at Lines 3-6 completes. In iteration $q' = j + 1 - n$, block $b_{\beta-1}$ is served to $H_{\beta-q'-1}$, hence,

$$\zeta_{\beta-1} = \bigcup_{p=0}^{n-\beta}\{H_{\beta+p-1}\} \bigcup_{p=0}^{q'-1}\{H_{\beta-1-p}\} \cup \{H_{\beta-1-q'}\}$$

and block $b_r, r \in \{0, 1, .., \beta-2\}$, is served to $H_{(r-q') \bmod n}$. Then, the state of block b_r,

$r \in \{0, ..., \beta - 1\}$, after the q'^{th} iteration is

$$\zeta_r = \bigcup_{p=0}^{n-\beta} \{H_{r+p}\} \bigcup_{p=0}^{q'-1} \{H_{(r-p) \bmod n}\} \cup \{H_{(r-q') \bmod n}\} = \bigcup_{p=0}^{n-\beta} \{H_{r+p}\} \bigcup_{p=0}^{q'} \{H_{(r-p) \bmod n}\}$$

■

Lemma 11. *During the execution of Algorithm 3, each host $H_i, i \in \{0, 1, .., n-1\}$, serves a block that it has already downloaded.*

Proof: In the *for* loop at Lines 3-6, during iteration $q = j + 1 - \beta, q \in \{1, .., n - \beta\}$, block b_r is served by H_{r+q-1}. It has it because after iteration $q - 1$,

$$\zeta_r = \bigcup_{p=0}^{q-1} \{H_{r+p}\},$$

which includes H_{r+q-1}. H_0 always serves b_0, if any, which it has from the above observation.

In the *for* loop at Lines 7-10, during iteration $q' = j + 1 - n, q' \in \{1, .., \beta - 1\}$, block $b_{\beta-1}$ is served by $H_{n-q'}$. It has it because after iteration $q' - 1$,

$$\zeta_{\beta-1} = \bigcup_{p=0}^{n-\beta} \{H_{\beta+p-1}\} \bigcup_{p=0}^{q'-1} \{H_{\beta-1-p}\} \cup \{H_{\beta-1-q'}\}$$

which includes $H_{n-q'}, \forall q' \in \{1, 2, .., \beta - 1\}$.

Block $b_r, r \in \{0, 1, .., \beta - 2\}$ is served by $H_{(r-(q'-1)) \bmod n}$. It has it because after iteration $q' - 1$

$$\zeta_r = \bigcup_{p=0}^{n-\beta} \{H_{r+p}\} \bigcup_{p=0}^{q'-1} \{H_{(r-p) \bmod n}\}$$

which includes $H_{(r-(q'-1)) \bmod n}$. Hence, the claim follows. ■

Lemma 12. *During the execution of Algorithm 3, a host is switched on (and off) at most thrice.*

Proof: In each of the for loops at Lines 1-2, 3-6, 7-10, a host is not switched on (resp. off) more than once, since indices i and j only increase in the loop. There are three such for loops, so a host can be switched on/off at most thrice in Algorithm 3. ■

Theorem 15. *After the termination of Algorithm 3 each host H_i, $i \in \{0, ..., n-1\}$ has received all the blocks $b_r \in \mathcal{B}$ with optimal energy $E(z_3) = \beta\left(\Delta_S + \sum_{i=0}^{n-1} \Delta_i\right) + (n-\beta)\min\{\Delta_S, \Delta_0\}$. Additionally, host i consumes exactly $\beta\Delta_i$ energy, except $H_{\min}$ that consumes $n\Delta_{min}$ energy, and no host has been switched on (and off) more than thrice.*

Proof: It follows from Lemma 10 that each host has received all the blocks. Then, the scheme is correct since each host serves blocks it has already downloaded (Lemma 11).

We need to bound now the energy consumed. Let us denote $\Delta_{\min} = \min\{\Delta_S, \Delta_0\}$. The energy consumed in the loop at Lines 1-2 is easily observed to be

$$E_1 = \beta\Delta_S + \sum_{i=0}^{\beta-1} \Delta_i \tag{A.6}$$

The energy consumed in the loop at Lines 3-6 is

$$E_2 = \sum_{j=\beta}^{n-1}\left(\Delta_{\min} + \Delta_{j+1-\beta} + \sum_{i=1}^{\beta-1}\Delta_{i+j+1-\beta}\right) = (n-\beta)\Delta_{\min} + \sum_{j=0}^{n-\beta-1}\sum_{i=0}^{\beta-1}\Delta_{i+j+1} \tag{A.7}$$

Finally, the energy consumed in the loop at Lines 7-10 is

$$E_3 = \sum_{j=n}^{n+\beta-2}\left(\Delta_{n+\beta-j-2} + \sum_{i=0}^{\beta-2}\Delta_{(n+i-j-1) \bmod n}\right) = \sum_{j=0}^{\beta-2}\sum_{i=0}^{\beta-1}\Delta_{(i-j-1) \bmod n} \tag{A.8}$$

Adding Equation A.6, A.7 and A.8, we get,

$$\begin{aligned}
E(z_3) &= E_1 + E_2 + E_3 \\
&= \beta\Delta_S + (n-\beta)\Delta_{\min} + \sum_{i=0}^{\beta-1}\left(\sum_{j=0}^{i}\Delta_j + \sum_{j=i+1}^{i+n-\beta}\Delta_j + \sum_{j=i+n-\beta+1}^{n-1}\Delta_j\right) \\
&= \beta\left(\Delta_S + \sum_{j=0}^{n-1}\Delta_j\right) + (n-\beta)\Delta_{\min},
\end{aligned}$$

which is optimal. This bound implies that all hosts are on exactly β slots, and hence consumes $\beta\Delta_i$ energy except $H_{\min}$ that is on for n slots consuming $n\Delta_{\min}$ energy. Hence, algorithm 3 is unfair to the host with minimum energy consumption. Finally, the bound of number of times a host is switched on/off is proven in Lemma 12.

■

A.2. Proof of Correctness and Performance of Algorithm 8

The proof of correctness of Algorithm 8 can be divided in essentially four parts. (We use the array abstraction for clarity.) The first claim is that, after the first loop (Lines 2-5), the diagonal of the first subarray has been filled. (I.e., $A_{ii} = 1, \forall i \in \{0, ..., n-1\}$.) This claim follows trivially by inspection. The second claim is that after the second loop (Lines 6-9), the top left corner position of each subarray has also been set to 1. (I.e., $A_{0j} = 1, \forall j \in \{0, n, 2n, .., (\lfloor \frac{\beta}{n} \rfloor - 1)n\}$.) This claim also follows by inspection.

The third claim is that, after the q^{th} iteration of the third loop (Lines 10-16), the whole

q^{th} subarray and the diagonal of the $(q+1)^{\text{th}}$ subarray have been set to 1 (and the blocks served by a host were available at the host for being served). This can be shown by induction on q, where the base case is the first claim above. In the induction step, the proof that the whole q^{th} subarray is set to 1 is similar to the proof of Algorithm 1. The proof that the diagonal of the $(q+1)^{\text{th}}$ subarray is set follows from the second claim above and Line 13 of the algorithm.

Finally, the fourth claim is that the process described in Line 45 completes the array. The proof of this claim is very similar to the proof of Algorithm 2.

Let us now compute the energy consumed by the scheme described by the algorithm. The first loop consumes energy $E_1 = 2n\Delta$, since the server is on n slots, while each client only one. The second loop consumes $E_2 = 2(\lfloor \beta/n \rfloor - 1)\Delta$, since in this loop both the server and the client H_0 are on $\lfloor \beta/n \rfloor - 1$ slots. The third loop uses energy

$$E_3 = \Delta \sum_{l=0}^{\lfloor \frac{\beta}{n} \rfloor - 2} \sum_{j=0}^{n-2} (n+1) = \Delta(\lfloor \frac{\beta}{n} \rfloor - 1)(n^2 - 1),$$

since in this loop all the hosts are on $(\lfloor \beta/n \rfloor - 2)(n-1)$ slots. Finally, the energy consumed by the process described in Line 45 is

$$E_4 = \Delta \left(\sum_{j=n}^{n+b-1} (n+1) + \sum_{j=n+b}^{n+b+n-2} n \right) = \Delta(b(n+1) + n(n-1)).$$

In this process no host is on more than $n+b$ slots. Adding up all these terms we compute the total energy as

$$E(z_4) = \Delta \left(n(\beta+1) + \left\lfloor \frac{\beta}{n} \right\rfloor + b - 1 \right).$$

Additionally, we bound the energy consumed by hosts as follows.

- The server is on for exactly β slots, consuming $\beta\Delta_S$ energy.
- The client H_0 is on for exactly $(\lfloor \beta/n \rfloor)n + b$ slots, consuming $((\lfloor \beta/n \rfloor)n + b)\Delta$ energy.
- And the rest of clients are on for exactly $\lfloor \frac{\beta}{n} \rfloor (n-1) + b + 1$ slots, consuming $(\lfloor \frac{\beta}{n} \rfloor (n-1) + b + 1)\Delta$ energy.

Thus, H_0 consumes $(\lfloor \frac{\beta}{n} \rfloor - 1)\Delta$ energy more than any other client. To prove that hosts switch on and off at most three times, the proof is analogous to that for previous algorithms. In the execution of Lines 2-9, all hosts switch on and off at most once except H_0, that switches on and off twice. In the rest of the algorithm, all clients are on until they finish downloading, and the server is switched off as soon as it serves all the blocks.

References

[1] J. R. Scott, A. P. Sokolov, S. Dutkiewicz, and C. E. Forest, "Probabilistic forecast for 21st century climate based on an ensemble of simulations using a business-as-usual scenario," in *AGU Fall Meeting Abstracts*, vol. 1, 2011, p. 0890.

[2] *Tackling Climate Change in the EU*. [Online]. Available: http://www.consilium.europa.eu/en/policies/climate-change/

[3] *Bureau of Energy Efficiency, Government of India*. [Online]. Available: https://beeindia.gov.in/

[4] *Office of Energy Efficiency & Renewable Energy*. [Online]. Available: http://energy.gov/eere/office-energy-efficiency-renewable-energy

[5] *Google Green Policy*. [Online]. Available: https://www.google.com/green/

[6] *Environmental Sustainability at Microsoft*. [Online]. Available: https://www.microsoft.com/about/csr/environment/

[7] *Environmental Sustainability at Microsoft*. [Online]. Available: http://www.cisco.com/web/about/citizenship/environment/energy_efficiency.html

[8] *ICT Codes of Conduct*. [Online]. Available: http://iet.jrc.ec.europa.eu/energyefficiency/ict-codes-conduct

[9] *ICT and CO_2 emissions*. [Online]. Available: http://www.parliament.uk/documents/post/postpn319.pdf

[10] K. J. Christensen, "The next frontier for communications networks: Power management," in *ITCom 2003*. International Society for Optics and Photonics, 2003, pp. 1–4.

[11] M. Gupta and S. Singh, "Greening of the Internet," in *SIGCOMM*, 2003.

[12] K. Christensen, C. Gunaratne, B. Nordman, and A. George, "The next frontier for communications networks: power management," *Computer Communications*, vol. 27, no. 18, pp. 1758–1770, 2004.

[13] G. Fettweis and E. Zimmermann, "Ict energy consumption-trends and challenges," in *Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications*, vol. 2, no. 4, 2008, p. 6.

[14] C. Gkantsidis, T. Karagiannis, and M. Vojnovic, "Planet scale software updates," in *ACM SIGCOMM*, 2006.

[15] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "Depsky: dependable and secure storage in a cloud-of-clouds," *ACM Transactions on Storage (TOS)*, vol. 9, no. 4, p. 12, 2013.

[16] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Computing*, vol. 18, no. 1, pp. 385–402, 2015.

[17] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, and X.-W. Wang, "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments," *Journal of computer science and technology*, vol. 27, no. 2, pp. 256–272, 2012.

[18] W. Li, Y. Yang, and D. Yuan, "A novel cost-effective dynamic data replication strategy for reliability in cloud data centres," in *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*. IEEE, 2011, pp. 496–502.

[19] N. Vasić and D. Kostić, "Energy-aware traffic engineering," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. ACM, 2010, pp. 169–178.

[20] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester, "Worldwide energy needs for ICT: The rise of power-aware networking," in *ANTS*, 2008.

[21] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ict electricity consumption from 2007 to 2012," *Computer Communications*, vol. 50, pp. 64–76, 2014.

[22] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *ACM SIGCOMM computer communication review*, vol. 39, no. 4. ACM, 2009, pp. 123–134.

[23] C. Gunaratne, K. Christensen, and B. Nordman, "Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed," *Int. J. Netw. Manag.*, vol. 15, pp. 297–310, September 2005.

[24] J. Arjona Aroca, A. Chatzipapas, A. Fernández Anta, and V. Mancuso, "A measurement-based analysis of the energy consumption of data center servers," in *Proceedings of the 5th international conference on Future energy systems, e-Energy 2014*. ACM, 2014, pp. 63–74.

[25] E. Goma, M. Canini, A. Lopez Toledo, N. Laoutaris, D. Kostić, P. Rodriguez, R. Stanojević, and P. Yagüe Valentin, "Insomnia in the access: or how to curb access network related energy consumption," in *ACM SIGCOMM*, 2011.

[26] G. Anastasi, S. Brienza, G. L. Re, and M. Ortolani, "Energy efficient protocol design," *Green Communications: Principles, Concepts and Practice*, pp. 339–360, 2015.

[27] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft, "Skilled in the art of being idle: reducing energy waste in networked systems," in *NSDI*, 2009.

[28] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *ACM SIGCOMM*, 2010.

[29] "Sandvine global Internet phenomena report 2h 2012," http://www.sandvine.com.

[30] A. Sucevic, L. Andrew, and T. Nguyen, "Powering down for energy efficient peer-to-peer file distribution," *ACM Sigmetrics Workshops, GreenMetrics*, 2011.

[31] L. L. Andrew, A. Sucevic, and T. T. Nguyen, "Balancing peer and server energy consumption in large peer-to-peer file distribution systems," in *IEEE Online Conference on Green Communications (GreenCom)*, 2011, pp. 76–81.

[32] H. Hlavacs, R. Weidlich, and T. Treutner, "Energy efficient peer-to-peer file sharing," *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1167–1188, 2012.

[33] M. Mehyar, W. Gu, S. Low, M. Effros, and T. Ho, "Optimal strategies for efficient peer-to-peer file sharing," in *Acoustics, Speech and Signal Processing, ICASSP*, vol. 4, 2007.

[34] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 223–244, 2011.

[35] J. Restrepo, C. Gruber, and C. Machuca, "Energy profile aware routing," in *Communications Workshops, IEEE ICC.*, 2009, pp. 1–5.

[36] M. Andrews, A. Fernández Anta, L. Zhang, and W. Zhao, “Routing for power minimization in the speed scaling model,” *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 1, pp. 285–294, 2012.

[37] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, “Somniloquy: augmenting network interfaces to reduce pc energy usage,” in *NSDI*, 2009, pp. 365–380.

[38] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, “A survey of green networking research,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 3–20, 2012.

[39] X. Wang, A. V. Vasilakos, M. Chen, Y. Liu, and T. T. Kwon, “A survey of green mobile networks: Opportunities and challenges,” *Mobile Networks and Applications*, vol. 17, no. 1, pp. 4–20, 2012.

[40] I. Kelenyi, A. Ludanyi, and J. K. Nurminen, “Bittorrent on mobile phones-energy efficiency of a distributed proxy solution,” in *Green Computing Conference, 2010 International*. IEEE, 2010, pp. 451–458.

[41] S. E. Cebeci, O. Ozkasap, and G. Anastasi, “Green proxy-based approaches for bittorrent,” in *Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on*. IEEE, 2014, pp. 153–156.

[42] G. Anastasi, I. Giannetti, and A. Passarella, “A bittorrent proxy for green Internet file sharing: Design and experimental evaluation,” *Computer Communications*, vol. 33, no. 7, pp. 794–802, 2010.

[43] S. Corigliano and P. Trunfio, “Exploiting sleep-and-wake strategies in the gnutella network,” in *Collaboration Technologies and Systems (CTS), 2014 International Conference on*. IEEE, 2014, pp. 406–412.

[44] T. Enokido, A. Aikebaier, and M. Takizawa, “A model for reducing power consumption in peer-to-peer systems,” *IEEE Systems Journal*, vol. 4, no. 2, pp. 221–229, 2010.

[45] ——, “Process allocation algorithms for saving power consumption in peer-to-peer systems,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2097–2105, 2011.

[46] P.-S. Tsang, X. Meng, and K.-S. Lui, “A novel grouping strategy for reducing average distribution time in P2P file sharing,” in *IEEE ICC*, 2010.

[47] A. Malatras, F. Peng, and B. Hirsbrunner, “Energy-efficient peer-to-peer networking and overlays,” *Handbook on Green Information and Communication Systems*, 2012.

[48] K. Park and P. Valduriez, "Energy efficient data access in mobile p2p networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp. 1619–1634, 2011.

[49] K. Verma, G. Rizzo, A. Fernández Anta, R. Cuevas Rumín, and A. Azcorra, "Greening file distribution: Centralized or distributed?" *ArXiv e-prints*, nov 2011.

[50] K. Verma, G. Rizzo, A. Fernández Anta, R. C. Rumín, A. Azcorra, S. Zaks, and A. García-Martínez, "Energy-optimal collaborative file distribution in wired networks," *Peer-to-Peer Networking and Applications*, pp. 1–20, 2016. [Online]. Available: http://dx.doi.org/10.1007/s12083-016-0453-4

[51] K. Verma, G. Rizzo, A. F. Anta, R. C. Rumin, and A. Azcorra, "Greening the internet: energy-optimal file distribution," in *Network Computing and Applications (NCA), 2012 11th IEEE International Symposium on*. IEEE, 2012, pp. 1–9.

[52] K. Verma, A. García-Martínez, and S. Agnihotri, "On the hardness of green file distribution and its practical applications," in *IEEE NCA 2016. Accepted for publication*.

[53] I. Giannetti, G. Anastasi, and M. Conti, "Energy-efficient p2p file sharing for residential bittorrent users," in *Computers and Communications (ISCC), 2012 IEEE Symposium on*. IEEE, 2012, pp. 000 524–000 529.

[54] M. Raj, K. Kant, and S. K. Das, "Energy adaptive mechanism for p2p file sharing protocols," in *European Conference on Parallel Processing*. Springer, 2012, pp. 89–99.

[55] A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy-efficient peer selection mechanism for bittorrent content distribution," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 1562–1567.

[56] P. Zhang and B. E. Helvik, "Towards green p2p: understanding the energy consumption in p2p under content pollution," in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010, pp. 332–337.

[57] ——, "Towards green p2p: Analysis of energy consumption in p2p and approaches to control," in *High Performance Computing and Simulation (HPCS), 2012 International Conference on*. IEEE, 2012, pp. 336–342.

[58] M. Forshaw and N. Thomas, "A novel approach to energy efficient content distribution with bittorrent," in *European Workshop on Performance Engineering*. Springer, 2012, pp. 188–196.

[59] J. Blackburn and K. Christensen, "A simulation study of a new green bittorrent," in *Communications Workshops, ICC*, 2009, pp. 1–6.

[60] A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Bittorrent content distribution in optical networks," *Journal of Lightwave Technology*, vol. 32, no. 21, pp. 3607–3623, 2014.

[61] E. Cem, E. Demirkaya, E. Esiner, B. Ozaydin, and O. Ozkasap, "Energy cost model for frequent item set discovery in unstructured p2p networks," in *Computer and Information Sciences II*. Springer, 2011, pp. 117–123.

[62] G. Jourjon, T. Rakotoarivelo, and M. Ott, "Models for an energy-efficient P2P delivery service," in *2010 18th IEEE Euromicro Conference on Parallel, Distributed and Network-based Processing*, 2010, pp. 348–355.

[63] Y.-J. Lee, J.-H. Jeong, H.-Y. Kim, and C.-H. Lee, "Energy-saving set-top box enhancement in bittorrent networks," in *2010 IEEE Network Operations and Management Symposium-NOMS 2010*. IEEE, 2010, pp. 809–812.

[64] P. Trunfio, "A two-layer model for improving the energy efficiency of file sharing peer-to-peer networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 13, pp. 3166–3183, 2015.

[65] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 211–222, 2012.

[66] S. Brienza, S. E. Cebeci, S. S. Masoumzadeh, H. Hlavacs, Ö. Özkasap, and G. Anastasi, "A survey on energy efficiency in p2p systems: File distribution, content streaming, and epidemics," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 36, 2016.

[67] I. Kelényi and J. K. Nurminen, "Energy aspects of peer cooperation measurements with a mobile dht system," in *ICC Workshops-2008 IEEE International Conference on Communications Workshops*. IEEE, 2008, pp. 164–168.

[68] I. Kelényi, Á. Ludányi, J. K. Nurminen, and I. Puustinen, "Energy-efficient mobile bittorrent with broadband router hosted proxies," in *Wireless and Mobile Networking Conference (WMNC), 2010 Third Joint IFIP*. IEEE, 2010, pp. 1–6.

[69] I. Kelényi, J. K. Nurminen, Á. Ludányi, and T. Lukovszki, "Modeling resource constrained bittorrent proxies for energy efficient mobile content sharing," *Peer-to-Peer Networking and Applications*, vol. 5, no. 2, pp. 163–177, 2012.

[70] I. Kelényi, Á. Ludányi, and J. K. Nurminen, "Distributed bittorrent proxy for energy efficient mobile content sharing," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*. IEEE, 2011, pp. 1–5.

[71] ——, "Energy-efficient bittorrent downloads to mobile phones through memory-limited proxies," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2011, pp. 715–719.

[72] I. Kelényi and J. K. Nurminen, "Cloudtorrent-energy-efficient bittorrent content sharing for mobile devices via cloud services," in *Proceedings of the 7th IEEE on Consumer Communications and Networking Conference (CCNC)*, vol. 1, 2010.

[73] M. Wichtlhuber, J. Rückert, D. Stingl, M. Schulz, and D. Hausheer, "Energy-efficient mobile p2p video streaming," in *2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P)*. IEEE, 2012, pp. 63–64.

[74] M.-H. Chen, C.-F. Chou, K.-H. Lee, and C.-Y. Chang, "On cooperative energy-efficient p2p live streaming system for mobile hotspots," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. IEEE, 2013, pp. 195–199.

[75] A. Chandrasekar, K. Chandrasekar, H. Ramasatagopan, and A. Rafica, "Smc: an energy conserving p2p file sharing model for mobile devices," in *Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM, 2012, pp. 66–73.

[76] S. Miyake and M. Bandai, "Energy-efficient mobile p2p communications based on context awareness," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*. IEEE, 2013, pp. 918–923.

[77] B. Barua, P. Karunakaran, H. Bagheri, and M. Katz, "Energy and delay efficient cooperative media content download," in *Wireless Days (WD), 2013 IFIP*. IEEE, 2013, pp. 1–4.

[78] M. Čagalj, J. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 172–182.

[79] C. Fragouli, J. Widmer, and J. Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice," in *IEEE Infocom*, 2006.

[80] J. Widmer, C. Fragouli, and J. Le Boudec, "Low-complexity energy-efficient broadcasting in wireless ad-hoc networks using network coding," in *Proc. Workshop on Network Coding, Theory, and Applications*, 2005.

[81] A. K.-H. Leung and Y.-K. Kwok, "On localized application-driven topology control for energy-efficient wireless peer-to-peer file sharing," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 66–80, 2008.

[82] J. Araujo, F. Giroire, J. Moulierac, Y. Liu, and R. Modrzejewski, “Energy efficient content distribution,” *The Computer Journal*, p. bxv095, 2015.

[83] V. Mathew, “Ph.D. Thesis: Energy-efficient content delivery networks,” 2015.

[84] V. Mathew, R. K. Sitaraman, and P. Shenoy, “Energy-aware load balancing in content delivery networks,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 954–962.

[85] ——, “Energy-efficient content delivery networks using cluster shutdown,” *Sustainable Computing: Informatics and Systems*, vol. 6, pp. 58–68, 2015.

[86] U. Lee, I. Rimac, D. Kilper, and V. Hilt, “Toward energy-efficient content dissemination,” *IEEE Network Magazine*, vol. 25, no. 2, pp. 14–19, 2011.

[87] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F. Westphal, “Energy trade-offs among content delivery architectures,” in *IEEE Telecommunications Internet and Media Techno Economics (CTTE)*, 2010, pp. 1–6.

[88] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, “Energy-efficient cloud computing,” *The computer journal*, vol. 53, no. 7, pp. 1045–1051, 2010.

[89] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, “Greening the Internet with nano data centers,” in *ACM CoNEXT*, 2009.

[90] R. Kumar and K. Ross, “Peer-assisted file distribution: The minimum distribution time,” in *IEEE Workshop on Hot Topics in Web Systems and Technologies (HOTWEB 06)*, 2006, pp. 1–11.

[91] M. Lingjun, P. Tsang, and K. Lui, “Improving file distribution performance by grouping in peer-to-peer networks,” *IEEE Transactions on Network and Service Management*, vol. 6, no. 3, pp. 149–162, 2009.

[92] T. Langner, C. Schindelhauer, and A. Souza, “Optimal file-distribution in heterogeneous and asymmetric storage networks,” *SOFSEM 2011: Theory and Practice of Computer Science*, pp. 368–381, 2011.

[93] S. Sanghavi, B. Hajek, and L. Massoulie, “Gossiping with multiple messages,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4640–4654, 2007.

[94] G. M. Ezovski, A. Tang, and L. L. Andrew, “Minimizing average finish time in P2P networks,” in *IEEE Infocom*, 2009.

[95] J. Mundinger, R. Weber, and G. Weiss, “Optimal scheduling of peer-to-peer file dissemination,” *Journal of Scheduling*, vol. 11, no. 2, pp. 105–120, 2008.

[96] ""in windows 7 use sleep to resume the os in 2 seconds","
http://news.softpedia.com/news/In-Windows-7-Use-Sleep-to-Resume-the-OS-in-2-Seconds-101290.shtml.

[97] B. Nordman and K. J. Christensen, "Greener PCs for the enterprise," *IT Professional*, vol. 11, no. 4, pp. 28–37, 2009.

[98] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[99] Intel, "Enhanced intel speedstep technology for the intel pentium m processor," Intel White Paper 301170-001, 2004.

[100] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," in *Power-aware computer systems*. Springer, 2004, pp. 165–180.

[101] W. L. Bircher and L. K. John, "Complete system power estimation using processor performance events," *Computers, IEEE Transactions on*, vol. 61, no. 4, pp. 563–577, 2012.

[102] ——, "Analysis of dynamic power management on multi-core processors," in *Proceedings of the 22nd annual international conference on Supercomputing*. ACM, 2008, pp. 327–338.

[103] *BitTorrent Algorithm*. [Online]. Available: http://www.cs.uiowa.edu/~ghosh/bittorrent.ppt

[104] "The real connection speeds for Internet users across the world (charts),"
http://royal.pingdom.com/2010/11/12/real-connection-speeds-for-internet-users-across-the-world/.

[105] P. Marciniak, N. Liogkas, A. Legout, and E. Kohler, "Small is not always beautiful," *arXiv preprint arXiv:0802.1015*, 2008.

[106] B. Cohen, "Incentives build robustness in bittorrent," in *Workshop on Economics of Peer-to-Peer systems*, vol. 6, 2003, pp. 68–72.